

Learning Tidy3D

Yankun (Alex) Meng

Lecture Outline

- Introduction to Tidy3D

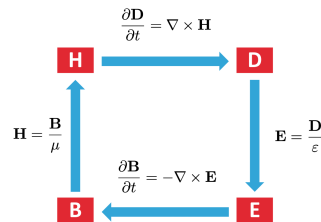
Introduction

FDTD method allows you to compute the evolution of electromagnetic field in the *time domain*.

Given some device $\epsilon(\vec{r})$ and an incident field or current source $\vec{J}(\vec{r}, t)$, internally, FDTD solves these maxwell's equations:

$$\nabla \times \vec{E}(\vec{r}, t) = -\mu_0 \frac{d\vec{H}(\vec{r}, t)}{dt} \quad (1)$$

$$\nabla \times \vec{H}(\vec{r}, t) = \epsilon(\vec{r})\epsilon_0 \frac{d\vec{E}(\vec{r}, t)}{dt} + \vec{J}(\vec{r}, t) \quad (2)$$



Field Update



The diagram is from [Simulation Documentation](#)

Starting Tidy3D

These software packages should be imported everytime you start a tidy3D project:

```
1 import tidy3d as td # Main package  
2 import tidy3d.web as web # Used to run the simulation  
3 import matplotlib.pyplot as plt # Used for plotting results  
4 import numpy as np # Used for numerical calculations
```

Before Simulation

Before the simulation, we have to define some key parameters of the electromagnetic waves that we will use across the simulation:

```
1 lambda_range = (1.1, 1.6)    # wavelength range ( $\mu m$ )
2 freqs = (td.C_0 / lambda_range[1], td.C_0 / lambda_range[0]) # frequency range
3 freq0 = np.mean(freqs)       # center frequency
4 lda0 = td.C_0 / freq0        # center wavelength
5 bandwidth = 0.38             # normalized bandwidth
6 freqw = bandwidth * (freqs[1] - freqs[0]) # bandwidth in Hz
```

Note: All numbers in tidy3d are in **microns** (μm)

Basic Workflow

Here's how to simulate something:

```
1 # Defining a simulation
2 simulation1 = td.Simulation(
3     # inputs
4 )
```

The 7 required inputs are: 1. Computational Domain Size 2. Grid Specifications (Discretization size) 3. Structures 4. Sources 5. Monitors 6. Run time 7. Boundary Condition Specification

We will introduce these 7 parameters by simulating a **huygen's metasurface**

Running a Simulation

The most basic way of running the simulation is using the *web* object we imported from tidy3d:

```
1 # Running a simulation
2 sim1_data = web.run(simulation1, task_name='any-unique-name', path='data/descriptive-name.
```

Simulation data is stored as an **HDF5 file** at the **file path** you specify.

1 Computational Domain Size

Size in x, y, and z directions.

```
1 p = 0.666 #nm → Periodicity
2 Lx, Ly, Lz = p, p, 2 * lda0
3 sim_size = [Lx, Ly, Lz]
4
5 # Defining a simulation
6 simulation1 = td.Simulation(
7     # A square computational domain
8     size = sim_size
9 )
```


2 Grid Specifications

Specifications for the simulation grid along each of the three directions.

```
1  # Define Grid size
2  spec = td.GridSpec.auto(min_steps_per_wvl=40, wavelength=lda0)
3
4  # Defining a simulation
5  simulation1 = td.Simulation(
6      # A square computational domain
7      size = (x, y, z),
8      grid_spec=spec,
9
10 )
```

- Typically, the size of a unit cell is $\frac{\lambda}{20}$

td.GridSpec contains many functions to help define the grid, the most commonly used are:

```
1 td.GridSpec.uniform(dl=grid_size)
2 td.GridSpec.auto(min_steps_per_wvl=40, wavelength=lda0)
```

- **uniform** – Use the same Uniform 1D grid along each of the three directions.
 - **dl** (float) – Grid size for uniform grid generation.
- **auto** – Use the same non-uniform grid along each of the three directions.
 - **min_steps_per_wvl**(ConstrainedFloatValue = 10.0) – Minimal number of steps per wavelength in each medium.
 - **wavelength** (float) – Wavelength to use for the step size and for dispersive media epsilon.

3 Structures

td.Structure is the meat of the simulation. It defines a physical object that interacts with the electromagnetic fields. The **structures** field is a list of **Structure** objects that you create.

```
1 # set up simulation
2 sim = td.Simulation(
3     size=sim_size,
4     grid_spec=spec,
5     structures=[superstrate, substrate, cylinder],
```

```

1  # set up simulation
2  td.Structure(
3      # inputs
4  )

```

A structure needs two inputs at least: – **geometry** (**td.Box**, **td.Cylinder**, **td.Sphere**, **td.TriangleMesh** (STL file), etc.) – **medium** – Mediums define the optical properties of the materials within the simulation. (e.g. **td.Medium**)

According to the paper on **huygen's metasurface**, I defined these four structures:

```

1  t = 2  # thickness of the substrate # THIS SHOULD BE CHANGED TO INFINITE
2  substrate = td.Structure(
3      geometry=td.Box(
4          center=(0,0,-t/2),
5          size=(td.inf,td.inf,t)
6      ),
7      medium=td.Medium(permittivity=1.45**2, name='oxide'),
8      name='substrate'
9  )

1  superstrate = td.Structure(
2      geometry=td.Box(
3          center=(0,0,t/2),
4          size=(td.inf,td.inf,t)
5      ),
6      medium=td.Medium(permittivity=1.45**2, name='glass')

```