Contents

1 FDTD **2**

2 MEEP **4**
2.1 Installation 4
2.2 Boundary Conditions 4
2.3 Countinuity 5
2.4 Material in Meep 5

3 Seeing things of a meep program. **6**
3.1 Simulation Domain 6
3.2 Geometry 6
3.3 Sources 6
3.4 Boundary layers 7
3.5 Resolution 7
3.6 Units 7
3.7 Sensor 7

4 Straight Wave-guide **9**

5 EMW in matter **10**
5.1 Transmittance and Reflectance on normal incidence 11

6 Plots from meep **13**
6.1 Transmittance for single Dielectric 13
6.1.1 Sensor at one place only 13
6.1.2 Sensor at many places 14
6.2 Transmittance for two different type of Dielectric 15
6.2.1 Sensor at one place only 15
6.2.2 Sensor at many places 16

7 Conclusion **17**

8 Codes uploaded on GitHub **17**

9 References **18**

Sixth semester project

Ravi Prakash Singh
1811122

Supervisor Dr.Satyaprasad P Senanayak

15/05/2021

Abstract

In this project, we focus to see the transmittance and reflectance of EMW in different wave-guide. For calculating this we used an open-source software package Meep, which is used for computational modelling, through the technique of FDTD. We learned the Conda a python package manager, different functions of NumPy and matplotlib, meep and plotted transmittance of EMW on a normal incidence to an interface.

Objective

- Study of Finite-difference time-domain (FDTD),a numerical analysis technique used for modeling computational electromagnetics,using Meep a free and open-source software package.
- Study the transmittance and loss through the dielectric when an EM Wave is passed through a wave-guide.

1 FDTD

A time-domain electromagnetic simulation simply evolves Maxwell's equations over time within some finite computational volume, essentially performing a kind of numerical experiment.Using this Transmittance Spectra, Reflectance Spectra, Resonant Modes and Frequencies , Field Patterns can be plotted.
FDTD is only one of several useful methods in computational electromagnetics. Actually is a widely used technique in which space is divided into a discrete grid and the fields are evolved in time using discrete time steps, as the grid and the time steps are made finer and finer, this becomes a closer and closer approximation for the true continuous equations, and one can

simulate many practical problems essentially exactly.
The accuracy of the modelling is decided by the grid size. This means the smaller the grid, more closer it is to reality so, the better it is.If the grid has some spatial resolution Δx, then our discrete time-step Δt is given by Δt=S Δx , where S is the Courant factor and must satisfy

$$S < n_{\min}/\sqrt{\#\text{ dimensions}}$$

where, $< n_{\min}$ is the minimum refractive index (usually 1), in order for the method to be stable (not diverge).

When Maxwell's differential equations are examined, it can be seen that the change in the E-field in time (the time derivative) is dependent on the change in the H-field across space (the curl). This results in the basic FDTD time-stepping relation that, at any point in space, the updated value of the E-field in time is dependent on the stored value of the E-field and the numerical curl of the local distribution of the H-field in space.
The H-field is time-stepped in a similar manner. At any point in space, the updated value of the H-field in time is dependent on the stored value of the H-field and the numerical curl of the local distribution of the E-field in space. Iterating the E-field and H-field updates results in a marching-in-time process wherein sampled-data analogs of the continuous electromagnetic waves under consideration propagate in a numerical grid stored in the computer memory.

To implement an FDTD solution of Maxwell's equations, a computational domain needs to be established, which is simply a physical region over which the simulation will be performed. The E and H fields are determined at every point in space within that computational domain. Then the material of each cell within the computational domain is specified (Free-space (air), metal, or dielectric) any material can be used as long as the permeability, permittivity, and conductivity can be specified.
Then a source is specified, which can be current on a wire, applied electric field or impinging plane wave. In the last case FDTD can be used to simulate EMW scattering from arbitrary shaped objects, planar periodic structures at various incident angles and photonic band structure of infinite periodic structures.
Since the E and H fields are determined directly, the output of the simulation is usually the E or H field at a point or a series of points within the computational domain but forward in time. Processing may be done on the E and H fields returned by the simulation. Data processing may also occur while the simulation is ongoing.

2 MEEP

Meep is a free and open-source software package for electromagnetics simulation via the finite-difference time-domain (FDTD) method spanning a broad range of applications.It is Free and open-source software under the GNU GPL, with complete scriptability via Python APIs.It is a precompiled binary packages of official releases and builds of the master branch via Conda.

2.1 Installation

Meep was originally developed as part of graduate research at MIT. The project has been under continuous development for nearly 20 years. It is currently maintained by Simpetus and the developer community on GitHub. The python API of MEEP can only be fulfilled as of now by using "ANACONDA". We need this because, Meep packages require a specific version of Python and its dependencies to run, so the strong environments features help us to keep things for Meep separated from other things we use.And basically Meep package is not available with any version of pip package installers.

2.2 Boundary Conditions

Since only a finite region of space can be simulated, the simulation must always be terminated with some boundary conditions. Three basic types of terminations are supported in Meep:

1:Bloch-periodic boundaries : With ordinary periodic boundaries in a cell of size L, the field components satisfy f(x+L)=f(x). Bloch periodicity is a generalization where f(x+L)=eikxLf(x) for some Bloch wavevector k. This can be used to solve for the modes of wave-guides, gratings, and so on.

2: Metallic walls : In this fields are simply forced to be zero on the boundaries, as if the cell were surrounded by a perfect metal (zero absorption, zero skin depth).

3: PML absorbing layers **We will be using this one mainly**:To simulate open boundary conditions, to absorb all waves incident on them, with no reflections. Perfectly matched layers (PML) is, strictly speaking, not a boundary condition — rather, it is a special absorbing material placed adjacent to the boundaries. PML is actually a fictitious (non-physical) material, designed to have zero reflections at its interface. Although PML is reflectionless in the theoretical continuous system, in the actual discretized system it has some small reflections which make it imperfect. For this rea-

son, one always gives the PML some finite thickness in which the absorption gradually turns on.

2.3 Countinuity

Although FDTD inherently uses discretized space and time, as much as possible Meep attempts to maintain the illusion that we are using a continuous system. At the beginning of the simulation, we specify the spatial resolution, but from that point onwards we generally work in continuous coordinates in our chosen units. See Units in Meep, above. Means, if we specify the dielectric function as a function of continuous x, or as a set of solid geometric objects such as a Sphere, Cylinder,or anything, then Meep is responsible for figuring out how they are to be represented on a discrete grid. Or if I want to specify a point Source, We simply specify the point x where we want the source to reside — Meep will figure out the closest grid points to x and add currents to those points, weighted according to their distance from x. If we change x continuously, the current in Meep will also change continuously by changing the weights.
In general, the philosophy of the Meep interface is pervasive interpolation, so that if we change any input continuously then the response of the Meep simulation will change continuously as well, so that it will converge as rapidly and as smoothly as possible to the continuous solution as we increase the spatial resolution.

2.4 Material in Meep

The material structure in Maxwell's equations is determined by the relative permittivity $\varepsilon(\mathbf{r})$ and permeability $\mu(\mathbf{r})$. However, $\varepsilon()$ is not only a function of position but in general, it also depends on frequency (material dispersion). It may also depend on the orientation of the field (anisotropy). Material dispersion, in turn, is generally associated with absorption loss in the material, or possibly gain. All of these effects can be simulated in Meep, with certain restrictions. Similarly for the relative permeability $\mu(\mathbf{r})$, for which dispersion, nonlinearity, and anisotropy are all supported as well. A materials library is available for Python and Scheme containing crystalline silicon (c-Si), amorphous silicon (a-Si) including the hydrogenated form, silicon dioxide (SiO2), indium tin oxide (ITO), alumina (Al2O3), gallium arsenide (GaAs) and many other. We have to just write- from meep.materials import "material name".

3 Seeing things of a meep program.

We will first import meep library, then we need to define a Simulation Domain i.e by setting computational cell.Then we will add a wave-guide and then a source.After adding the source we will require to set the environment of surrounding by setting the boundary layers.
After all these we need to set some analysis function and then use it to show in simulations. So we need to define the resolution and use numpy and mathplotlib to address the data management and visualization works.

3.1 Simulation Domain

We can specify each of the simulation objects starting with the computational cell. We can put a source at one end and watch the fields propagate down the wave-guide in the x direction, use a cell of length in the x direction to give it some distance to propagate. In the y direction, we just need enough room so that the boundaries do not affect the wave-guide mode.All length measurement in meep are in m by default.

3.2 Geometry

We need to add the wave-guide.The device structure is specified by a set of GeometricObjects stored in the geometry object. The wave-guide is specified by a different shape, size and material. By default, any place where there are no objects there is air (=1), although this can be changed by setting the default material variable.

Material [Medium class or function]- The material that the object is made of (usually some sort of dielectric). Uses default Medium. If a function is supplied, it must take one argument and return a Python Medium.

Epsilon func [function]- A function that takes one argument (a Vector3) and returns the dielectric constant at that point. Can be used instead of material. Default is None.

Center [Vector3] - Center point of the object. Defaults to $(0, 0, 0)$.

3.3 Sources

We can set the different type of sources in meep.Continuous-wave (CW) source is proportional to $\exp(-i\omega t)$, possibly with a smooth (exponential/tanh) turn-on/turn-off. In practice, the CW source never produces an exact single-frequency response.CW source gives an output of constant frequency.The

GaussianSource is parameterized by a center frequency and a frequency width (the width of the Gaussian spectrum) , at which it radiates.

3.4 Boundary layers

As for boundary conditions, we want to add absorbing boundaries around our cell.Absorbing boundaries in Meep are handled by perfectly matched layers (PML) — which aren't really a boundary condition at all, but rather a fictitious absorbing material added around the edges of the cell.
PML layers is a list of PML objects, we may have more than one PML object if we want PML layers only on certain sides of the cell. For example, mp.PML(thickness=1.0,direction=mp.X,side=mp.high) specifies a PML layer on only the +x side. An important point: the PML layer is inside the cell, overlapping whatever objects you have there. So, in this case our PML overlaps our wave-guide, which is what we want so that it will properly absorb wave-guide modes.

3.5 Resolution

Meep will discretize this structure in space and time, and that is specified by a single variable, resolution, that gives the number of pixels per distance unit. We'll set this resolution to 10 pixels/m by setting resolution =10, which corresponds to around 67 pixels/wavelength, or around 20 pixels/wavelength in the high-index material. In general, at least 8 pixels/wavelength in the highest dielectric is a good idea.

3.6 Units

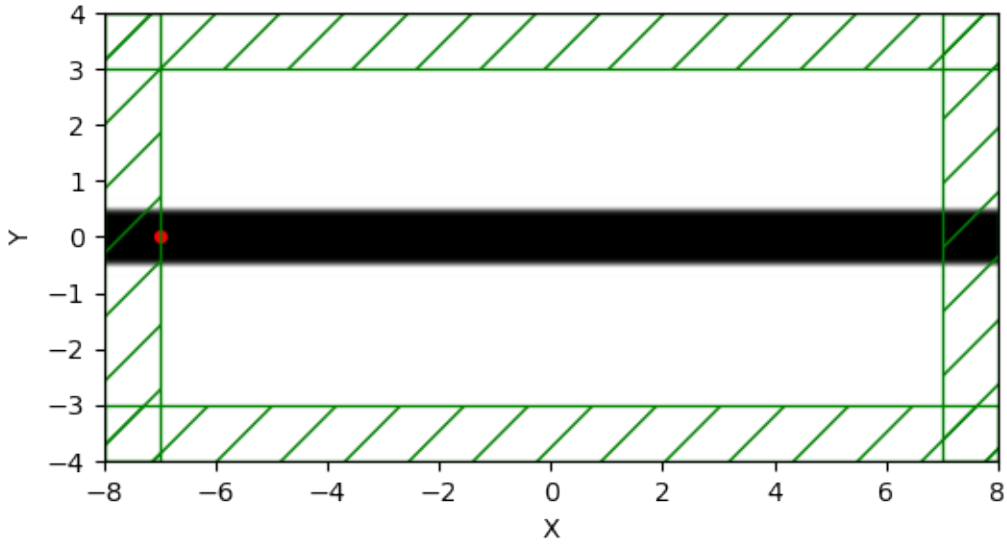
In Meep units, frequency is specified in units of $2\pi c$, which is equivalent to the inverse of the vacuum wavelength. Thus, 0.15 corresponds to a vacuum wavelength of about $1/0.15 = 6.67\mu$m, or a wavelength of about 2μm in the $\varepsilon = 12$ material.
All length measurement in meep are in m by default.

3.7 Sensor

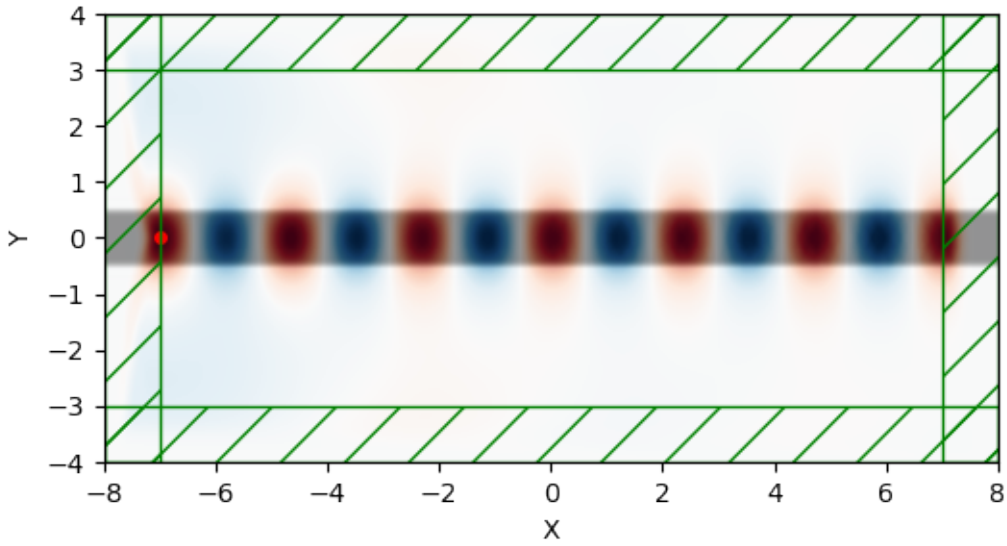
We need sensors to calculate the energy passing through, and we have to specify where we want Meep to compute the flux and at what frequencies.This must be done after specifying the Simulation object, which contains the geometry, sources, resolution and other things. The flux is the integral of the Poynting vector over the specified FluxRegion. It only integrates one component of the Poynting vector, and the direction property specifies which component. In our case, since the FluxRegion is a line, the direction is its normal by default.

We compute the fluxes through a line segment equal to the width of the waveguide, located at the desired place in the waveguide. We only compute fluxes for frequencies within our source bandwidth. This is important because, far outside the bandwidth, the spectral power is so low that numerical errors make the computed fluxes useless. All of this is accomplished with three commands which use the raw simulation data:
"$FluxRegion$" to set the Sensor geometry and location.
"add_flux" to set the frequency at which we want to make sensor active.
"get_fluxes" to Save flux energy in an array.

4 Straight Wave-guide



The black portion here is the wave-guide. The rectangle area of (4 to -4) X (-8 to 8) is our computational source. The green colour shadow area is the pml layer. We take source as Continuous Source of frequency=0.15 and z -component centered at (-7,0) as a point source.



In this plot we have Ez component plotted over the wave-guide here we use the "RdBu" colormap which goes from dark red (negative) to white (zero) to dark blue (positive).

5 EMW in matter

Inside matter, but in regions where there is no free charge or free current, Maxwell's equations become

(i) $\nabla \cdot \mathbf{D} = 0$ (iii) $\boldsymbol{\nabla} \times \mathbf{E} = -\frac{\partial \mathbf{B}}{\partial t}$, (ii) $\boldsymbol{\nabla} \cdot \mathbf{B} = 0$, (iv) $\boldsymbol{\nabla} \times \mathbf{H} = \frac{\partial \mathbf{D}}{\partial t}$.

If the medium is linear

$$\mathbf{D} = \epsilon \mathbf{E}, \quad \mathbf{H} = \frac{1}{\mu}\mathbf{B}$$

and homogeneous (so ϵ and μ do not vary from point to point), they become

(i) $\boldsymbol{\nabla} \cdot \mathbf{E} = 0$ (iii) $\boldsymbol{\nabla} \times \mathbf{E} = -\frac{\partial \mathbf{B}}{\partial t}$, (ii) $\boldsymbol{\nabla} \cdot \mathbf{B} = 0$, (iv) $\boldsymbol{\nabla} \times \mathbf{B} = \mu\epsilon\frac{\partial \mathbf{E}}{\partial t}$

Evidently electromagnetic waves propagate through a linear homogeneous medium at a speed

$$v = \frac{1}{\sqrt{\epsilon\mu}} = \frac{c}{n}$$

where,

$$n \equiv \sqrt{\frac{\epsilon\mu}{\epsilon_0\mu_0}}$$

is the index of refraction of the substance. For most materials, μ is very close to μ_0, so

$$n \cong \sqrt{\epsilon_r},$$

where ϵ_r is the dielectric constant. Since ϵ_r is almost always greater than 1, EMW travels more slowly through matter.

$\epsilon_0 \to \epsilon$, $\mu_0 \to \mu$, and hence $c \to v$. The energy density is

$$u = \frac{1}{2}\left(\epsilon E^2 + \frac{1}{\mu}B^2\right)$$

and the Poynting vector is

$$\mathbf{S} = \frac{1}{\mu}(\mathbf{E} \times \mathbf{B})$$

For monochromatic plane waves, the frequency and wave number are related by $\omega = kv$, the amplitude of $\mathbf{B}$ is $1/v$ times the amplitude of $\mathbf{E}$, and the intensity is

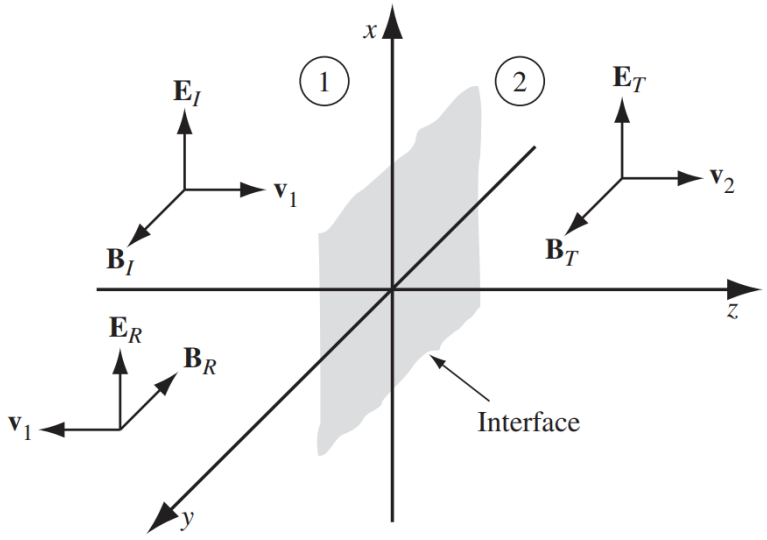
$$\mathbf{I} = \frac{\mathbf{1}}{\mathbf{2}}\epsilon \mathbf{v} \mathbf{E_0^2}$$

(i) $\epsilon_1 E_1^\perp = \epsilon_2 E_2^\perp$, (iii) $\mathbf{E}_1^\parallel = \mathbf{E}_2^\parallel$, (ii) $B_1^\perp = B_2^\perp$, (iv) $\frac{1}{\mu_1}\mathbf{B}_1^\parallel = \frac{1}{\mu_2}\mathbf{B}_2^\parallel$.

These equations relate the electric and magnetic fields just to the left and just to the right of the interface between two linear media.

5.1 Transmittance and Reflectance on normal incidence

Suppose the xy plane forms the boundary between two linear media. A plane wave of frequency ω, traveling in the z direction and polarized in the x direction, approaches the interface from the left



$$\tilde{\mathbf{E}}_I(z,t) = \tilde{E}_{0_I} e^{i(k_1 z - \omega t)} \hat{\mathbf{x}}$$

$$\tilde{\mathbf{B}}_I(z,t) = \frac{1}{v_1} \tilde{E}_{0_I} e^{i(k_1 z - \omega t)} \hat{\mathbf{y}}$$

It gives rise to a reflected wave

$$\tilde{\mathbf{E}}_R(z,t) = \tilde{E}_{0_R} e^{i(-k_1 z - \omega t)} \hat{\mathbf{x}}$$

$$\tilde{\mathbf{B}}_R(z,t) = -\frac{1}{v_1} \tilde{E}_{0_R} e^{i(-k_1 z - \omega t)} \hat{\mathbf{y}}$$

which travels back to the left in medium (1), and a transmitted wave

$$\tilde{\mathbf{E}}_T(z,t) = \tilde{E}_{0_T} e^{i(k_2 z - \omega t)} \hat{\mathbf{x}},$$

$$\tilde{\mathbf{B}}_T(z,t) = \frac{1}{v_2} \tilde{E}_{0_T} e^{i(k_2 z - \omega t)} \hat{\mathbf{y}}$$

which continues on to the right in medium (2). The Poynting vector aims in the direction of propagation.

At $z = 0$, the combined fields on the left, $\tilde{\mathbf{E}}_I + \tilde{\mathbf{E}}_R$ and $\tilde{\mathbf{B}}_I + \tilde{\mathbf{B}}_R$, must join the fields on the right, $\tilde{\mathbf{E}}_T$ and $\tilde{\mathbf{B}}_T$, in accordance with the boundary conditions

$$\epsilon_1 E_1^{\perp} = \epsilon_2 E_2^{\perp}$$

$$B_1^{\perp} = B_2^{\perp}$$

In this case there are no components perpendicular to the surface, so the above two equations are trivial.
Then,

$$\mathbf{E}_1^{\parallel} = \mathbf{E}_2^{\parallel}$$

Gives $\tilde{E}_{0_I} + \tilde{E}_{0_R} = \tilde{E}_{0_T}$ ——(1)

$$\frac{1}{\mu_1}\mathbf{B}_1^{\parallel} = \frac{1}{\mu_2}\mathbf{B}_2^{\parallel}$$

gives, $\frac{1}{\mu_1}\left(\frac{1}{v_1}\tilde{E}_{0_I} - \frac{1}{v_1}\tilde{E}_{0_R}\right) = \frac{1}{\mu_2}\left(\frac{1}{v_2}\tilde{E}_{0_T}\right)$
which is $\tilde{E}_{0_I} - \tilde{E}_{0_R} = \beta\tilde{E}_{0_T}$——(2)
where,

$$\beta \equiv \frac{\mu_1 v_1}{\mu_2 v_2} = \frac{\mu_1 n_2}{\mu_2 n_1}$$

In terms of velocity, v_1 in medium 1 and v_2 in medium 2 (using (1) and (2))

$$E_{0_T} = \left(\frac{2v_2}{v_2 + v_1}\right) E_{0_l}$$

In terms of refractive index, n_1 in medium 1 and n_2 in medium 2

$$E_{0_T} = \left(\frac{2n_1}{n_1 + n_2}\right) E_{0_I}$$

T (transmission coefficient) is the ratio of the transmitted intensity to the incident intensity is

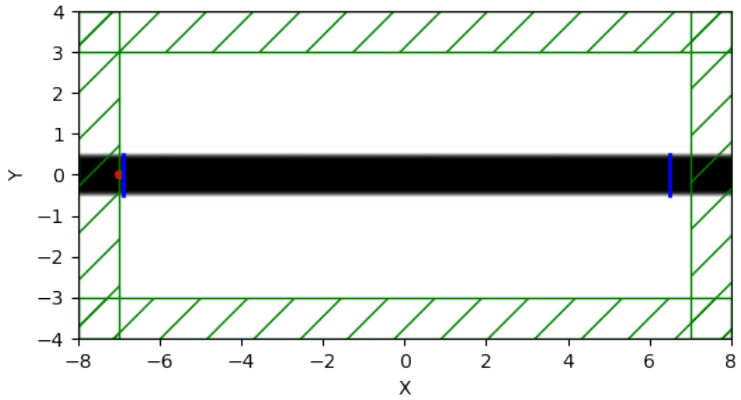
$$T \equiv \frac{I_T}{I_I} = \frac{\epsilon_2 v_2}{\epsilon_1 v_1}\left(\frac{E_{0_T}}{E_{0_I}}\right)^2 = \frac{4n_1 n_2}{(n_1 + n_2)^2}$$

R the reflection coefficient is 1-T

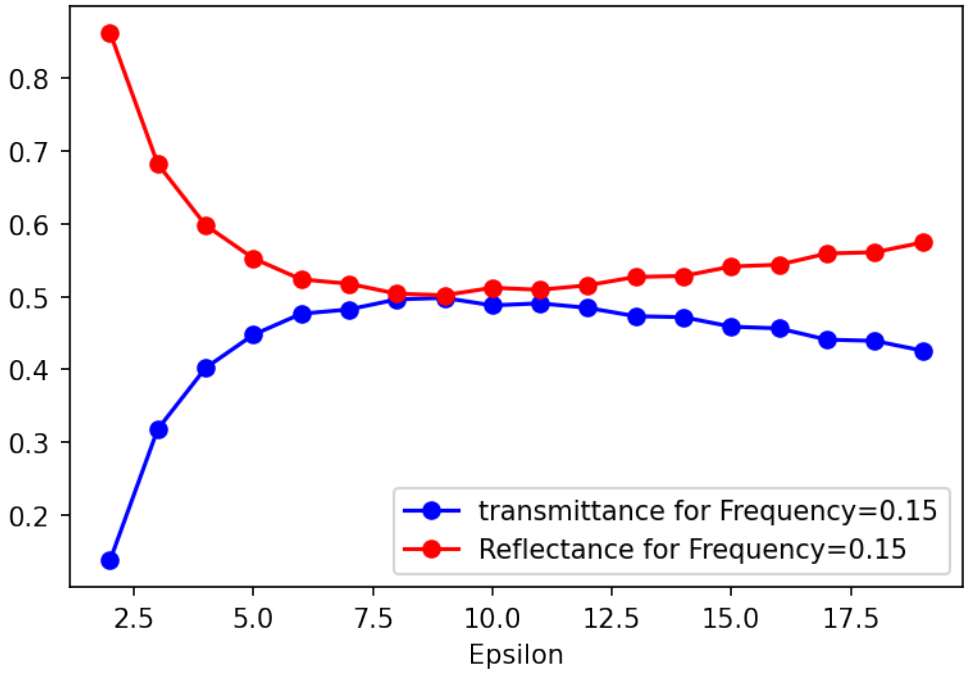
6 Plots from meep

6.1 Transmittance for single Dielectric

6.1.1 Sensor at one place only

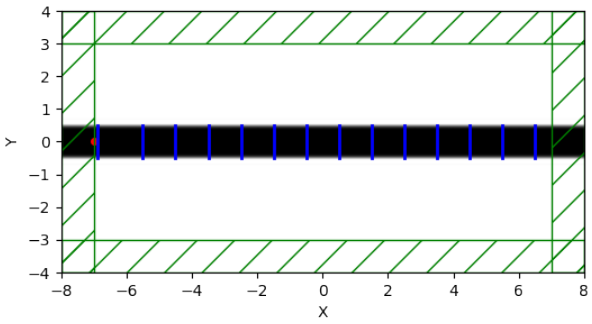


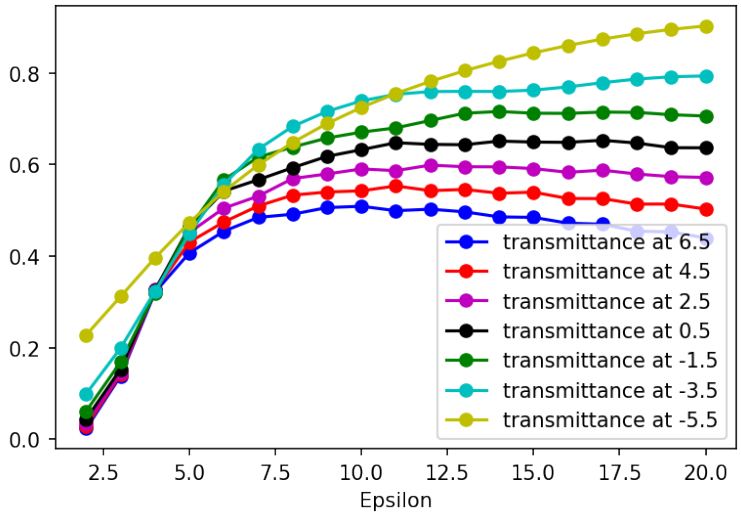
Here the blue lines are the sensor at which we can calculate (Power) energy passing through.Basically T=(Power at sensor 2)/(Power at sensor1). So we have plotted the graph for transmittance and reflectance for a single uncut material at different dielectric value of whole wave-guide material.

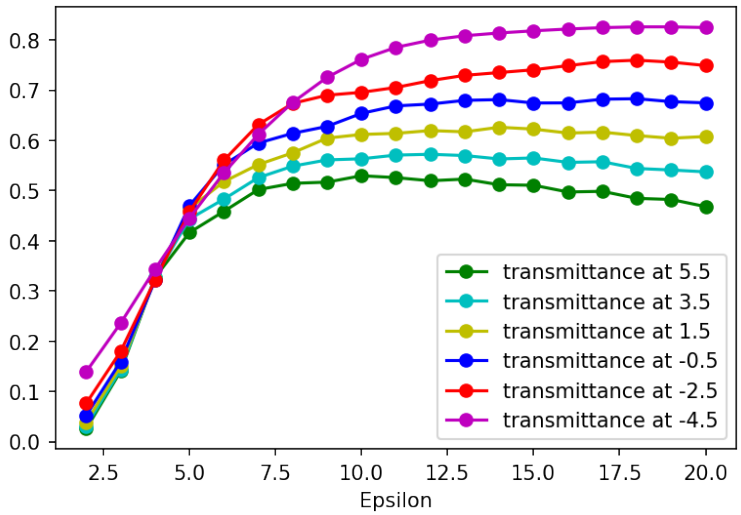


6.1.2 Sensor at many places

Dielectric material and conditions remain same here also just the no. of sensors are increased so that we can calculate transmittance at different places in the wave-guide.

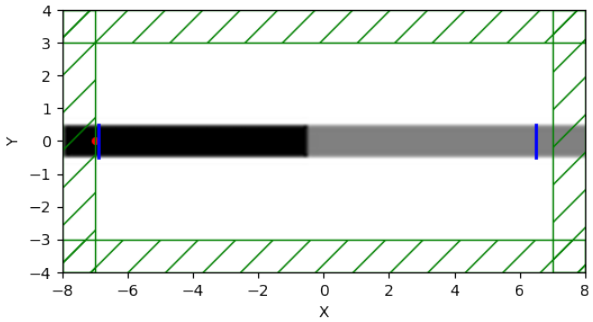




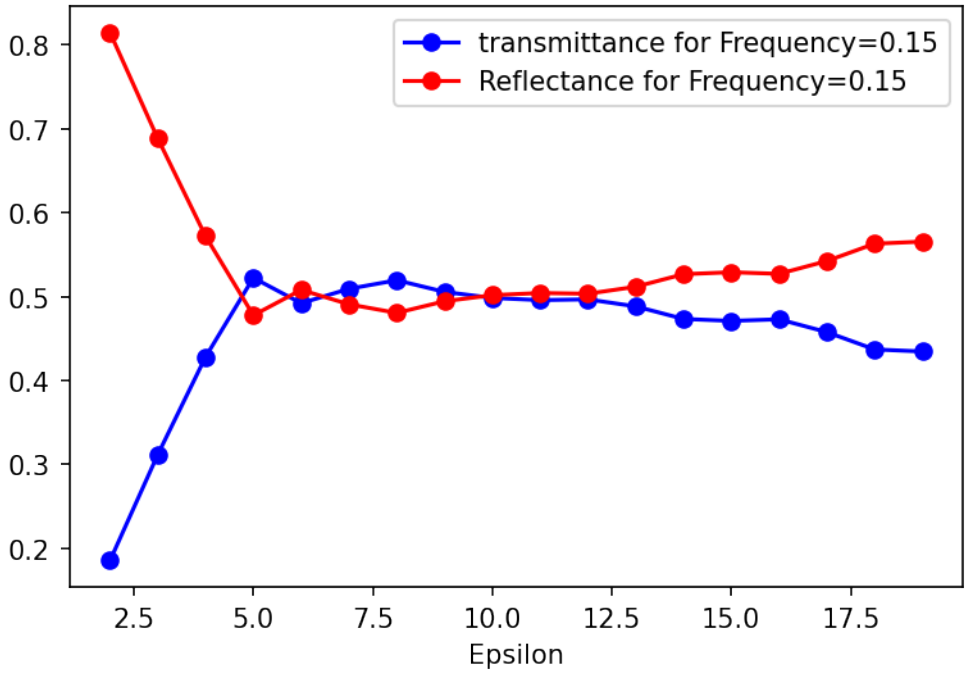


6.2 Transmittance for two different type of Dielectric

6.2.1 Sensor at one place only

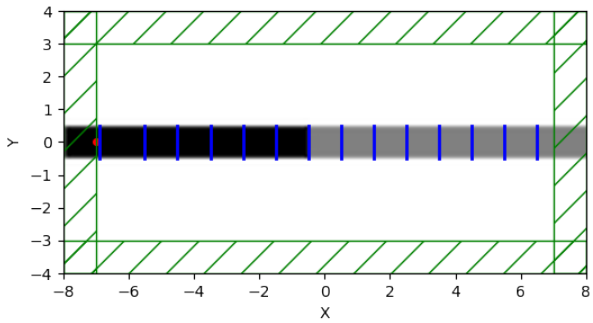


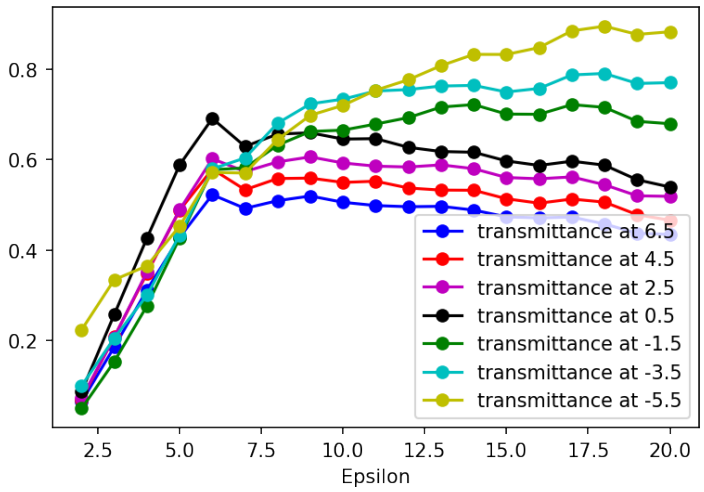
Here the blue lines are the sensor at which we can calculate (Power) energy passing through.Basically T=(Power at sensor 2)/(Power at sensor1). So we have plotted the graph for transmittance and reflectance for material divided into two parts, the left part is fixed at $\epsilon = 10$ while the ϵ of right is varied from 1 to 20 and transmittance is measured.

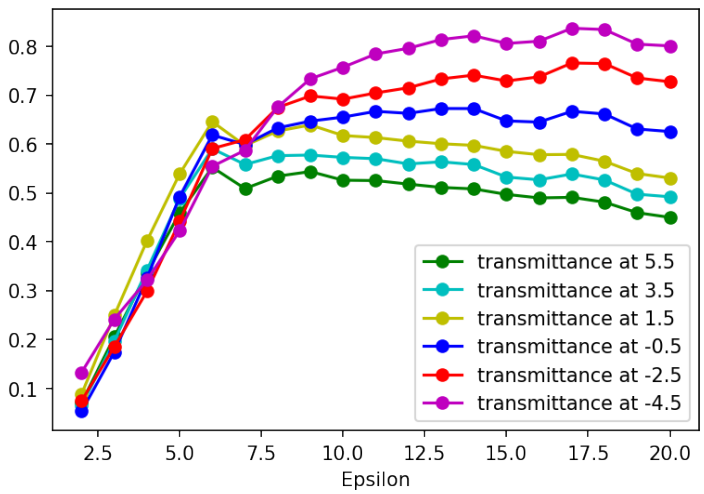


6.2.2 Sensor at many places

Dielectric material and conditions remain same here also just the no. of sensors are increased so that we can calculate transmittance at different places in the wave-guide.







7 Conclusion

6.1.1 →
A-→The loss in transmittance in case of single medium (Dielectric) is due to electric field hitting wave guide surface at more than critical angle and hence get out of the wave-guide.
B-→The Transmittance starts with very less and goes on increasing and then saturates after some values of epsilon because after some increase in value the critical angle is high enough and any further further increase in critical angle will however increase the reflection but the path to follow will be so large that they will not reach the other sensor.
6.1.2 →
A→As the sensor closer to the sensor 1 is choser the transmittance value increases because the loss due moving in dielectric is getting less and less.The major reason of transmittance for the closest sensor not being 1 is the loss due to hitting above critical angle.This can be seen very clear as the saturation is being acheived very late for the closer sensor in compare with the away one.
B→The pml layers used are also not complete absorbing as the EMW incident non perpendicular to surface can be reflected back so they may be a reason in the sudden jumps in the graph.
6.2.1→
A→ In this case all the points of (6.1.1.A and B) holds but one majoy point is added that is Transmittance on normal incidence on a interface of different medium ($T = \frac{\epsilon_2 v_2}{\epsilon_1 v_1}\left(\frac{E_{0_T}}{E_{0_I}}\right)^2 = \frac{4n_1n_2}{(n_1+n_2)^2}$).
B → So to calculate the Transmittance on normal incidence means loss due to only reflection we should add the loss due to critical angle and heating to this.

C→The pml layers used are also not complete absorbing as the EMW incident non perpendicular to surface can be reflected back so they may be a reason in the sudden jumps in the graph.

8 Codes uploaded on GitHub

Link of all my codes in (https://github.com/1raviprakash/6th-semester-project)

9 References

- http://optics.hanyang.ac.kr/ shsong/Chapter
- https://www.google.com/search?q=meep+documentationoq=meeaqs=chrome.0.69i
- https://meep.readthedocs.io/en/latest/Perfectly$_{M}atched_{L}ayerIntroductiontoElectrodynamic$

- https://conda.io