Finite-Difference-Time-Domain Simulation of Ultrafast Experiments

Alpha Ma

Department of Physics and Astronomy, Macalester College, 1600 Grand Ave, Saint Paul, MN

Abstract

The Finite-Difference-Time-Domain (FDTD) method is a numerical method that calculates electric fields or magnetic fields by interleaving them in space and time. Using a python package called "MEEP", I was able to write optical simulations of ultrafast experiments, especially the Terahertz Pump-Probe experiments. The goal of this project was to use FDTD simulation to measure the transmission of an electro-magnetic pulse passing through a thin film of conducting material on a dielectric substrate in order to study the characteristic conductivity of potential solar cell materials.

Keywords: FDTD; "MEEP"; pump-probe experiments; amplitude transmission; simulations

Introduction

Ultrafast conductivity measurement experiments are well applied by Physicists and engineers on solar cell materials to study their performance and efficiency. Conducting these experiments can often be very expensive in terms of time and resources. Numerical methods have been developed to solve Maxwell's equations to simulate optical experiments, and they can be implemented using computer programs. The Finite-difference time-domain (FDTD) method was developed by Yee in the 1960s to solve Maxwell's curl equations on grids. Using a python package called “MEEP” which implements the FDTD method, we can write computer programs to simulate ultrafast experiments. We studied pump-probe experiments where a pump pulse is sent to a solar cell material and excites the sample, and the probe pulse which is sent to the sample femtoseconds to nanoseconds later transmits through the material and measures the change in the properties of the sample due to excitation by the pump. The amplitude transmission spectrum of the probe-pulse can be measured by a detector. The simulation results can be used to compare with the experimental data collected from the THZ lab in Macalester College, MN shown in Figure.1. Once the accuracy and the consistency of the simulations reach an agreeable level, the simulation results can be used to check the consistency of the experimental results, and they can also be used to study the conductivity profile of solar cell materials such as GaAsN or the newly developed perovskite solar cells.

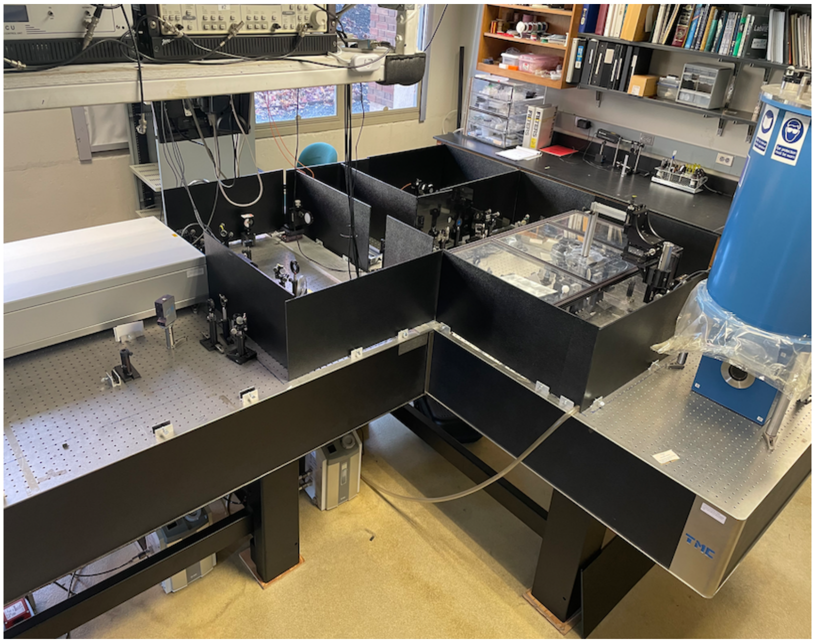
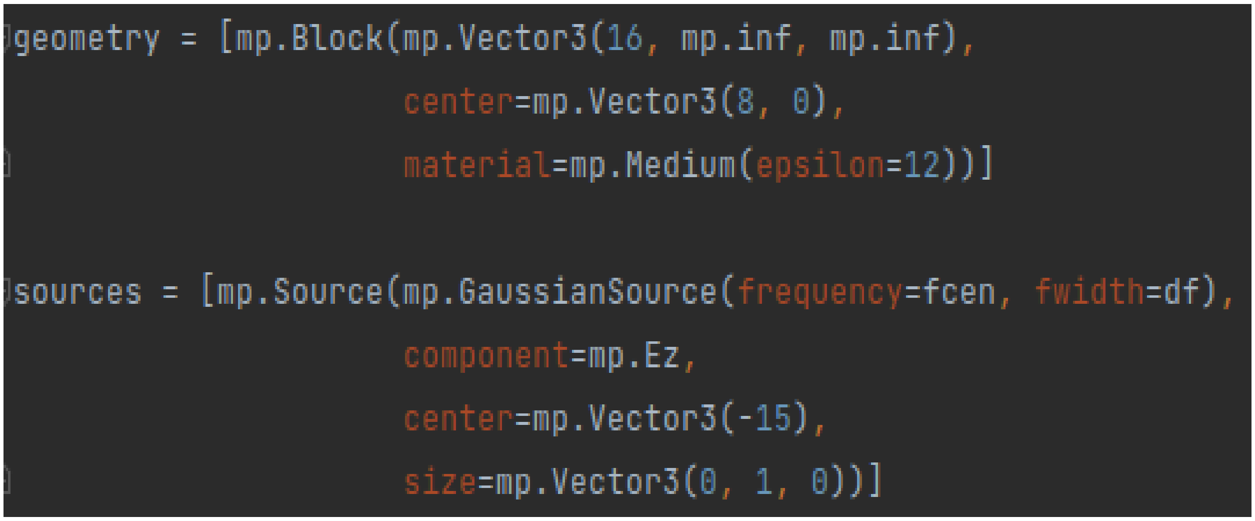


Figure.1: Femto XL-500 laser, THz Lab, Macalester College, MN

In addition to the computer simulation, it is also possible to obtain the transmission spectrum using a two-dimensional analytical model of the scenario. The 2-D treatment is derived from the Huygen-Fresnel principle and it provides an analytical expression to calculate the transmission spectrum. Using Mathematica, we are able to obtain the spectrum to compare with the simulation spectrum. The 2-D analytical treatment can check the accuracy of the simulation results before comparing the simulation spectrum with the experimental data.

FDTD Simulations Using "MEEP"

The Finite-Difference-Time-Domain method is implemented by an open-source python package called "MEEP". MEEP is designed to simulate ultrafast experiments and it allows users to exploit symmetries in order to reduce the simulation dimension. A "MEEP" simulation requires definitions of cell size, resolution(how many pixels in a cell), source profile, and dielectric profile. Typical codes for defining the dielectric profile are shown in Figure.2.



```
geometry = [mp.Block(mp.Vector3(16, mp.inf, mp.inf),
                     center=mp.Vector3(8, 0),
                     material=mp.Medium(epsilon=12))]

sources = [mp.Source(mp.GaussianSource(frequency=fcen, fwidth=df),
                     component=mp.Ez,
                     center=mp.Vector3(-15),
                     size=mp.Vector3(0, 1, 0))]
```

Figure.2: A chunk of "MEEP" codes to define the geometry of the dielectric material and the source profile.

Appendix A contains more detailed codes for constructing a simulation and manipulating the outputs. In order to retain accuracy but not cost too much time, the simulations were constructed two-dimensionally by exploiting the symmetry in the problem. In order to check the accuracy of "MEEP" simulations, I constructed a simulation with a material that has a Drude-conductivity profile. The Drude-conductivity can be described using the following equation:

$$\sigma = \sigma_0/(1 - i\omega\tau) \tag{1}$$

The Drude model conductivity assumes the charge carriers to be free particles in the material until they scatter. The charge carriers have an average scattering time $\boldsymbol{\tau}$, and after each scattering, charge carriers lose all momentum. In equation (1), σ_0 is the DC-conductivity which is a constant; ω is the frequency with units (rad/s); $\boldsymbol{\tau}$ is the mean scattering time with units (s). The Drude-model is a simple but accurate model to describe conducting materials with frequency-dependent complex conductivity. In order to see the performance of "MEEP" simulations, I obtained the simulation transmission spectrum and compared it with a 1-D analytical spectrum. (Shown in Figure.3). From Figure.3, we can see that the FDTD simulation result agrees reasonably well with the 1-D analytical spectrum, except with some discrepancies at higher or lower frequencies.

The reason why there are discrepancies at the ends of the frequency range is that "MEEP" performs a discrete Fourier transformation on the electric fields in the space within a range of frequencies. The edge discrepancies are unavoidable if the frequency range is about twice the center frequency. In general, it is still a fairly accurate result for a two-dimensional simulation.

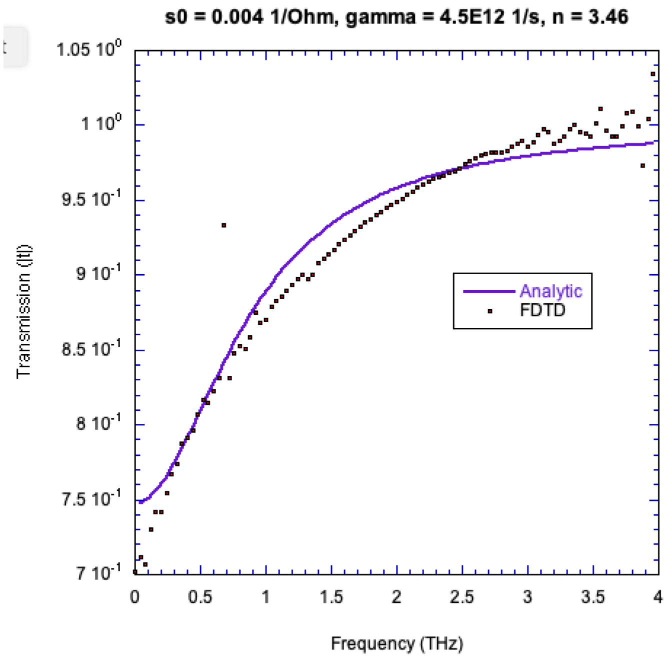


Figure.3: A graph that compares the transmission spectra of a Drude-model material between Analytical and FDTF simulation results. γ(gamma) = 1/$\boldsymbol{\tau}$

The Pump-Probe Simulation

Now since we have an understanding of how accurate the FDTD simulations are, it is necessary to test it out by constructing a simulation for the actual pump-probe experiment conducted in the THz lab (shown in Figure.4). The XL-500 laser sends out an 800nm pump pulse and a following THz probe pulse. The two pulses travel in different paths and the displacement in the paths leads to a difference between the arrival time of the pump-pulse and the probe-pulse that can be varied from zero to two nanoseconds with femtosecond accuracy. There is a 1mm radius aperture attached to the sample, and the pump pulse arriving at the sample first excites a position-varying conductivity in the area of the aperture on the sample. The late-arriving THz probe-pulse then can transmit through the aperture and is captured by a detector in order to measure the amplitude transmission spectrum. A “MEEP” simulation of the described pump-probe experiment is shown in Figure.5. The position-varying conductivity can be described by equation (2).

$$\sigma(y) = \sigma_0 e^{-y^2/w^2}/(1 - i\omega/\gamma) \qquad (2)$$

(2) where w is the radius of the aperture, y is the distance from the center of the aperture to the point on the sample.

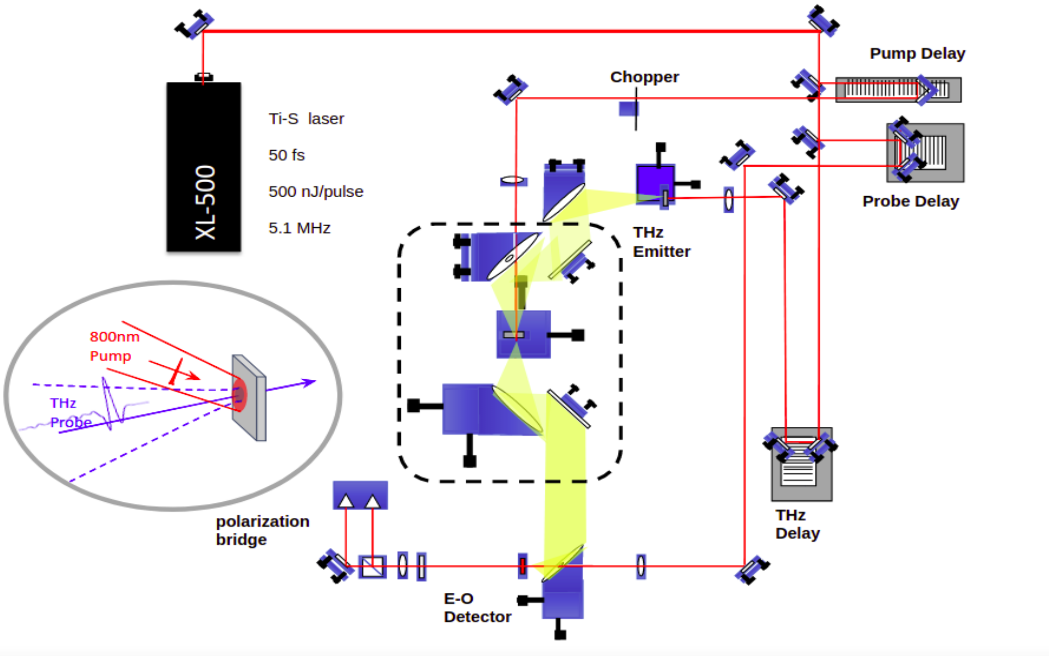


Figure.4: A diagram of the pump-probe experiment. The pump-pulse and probe-pulse are displaced so that they arrived at the sample in a femtoseconds difference.

The actual experiment has the following parameters:

Parameters	**Value**
w(radius of the aperture)	0.5 (mm)
γ	$4\cdot10^{12}$ 1/s
ε_r(dielectric constant)	12
σ_0	0.004 (1/Ω)

Table.1: parameters of the pump-probe experiment performed in the THz lab.

The amplitude transmission spectrum was then obtained from the simulation, shown in Figure.6.

Figure.5: The output gif of a "MEEP" simulation of the pump-probe experiment with a plane wave traveling to the sample attached to a dielectric medium. The codes for the simulation can be found in Appendix B.

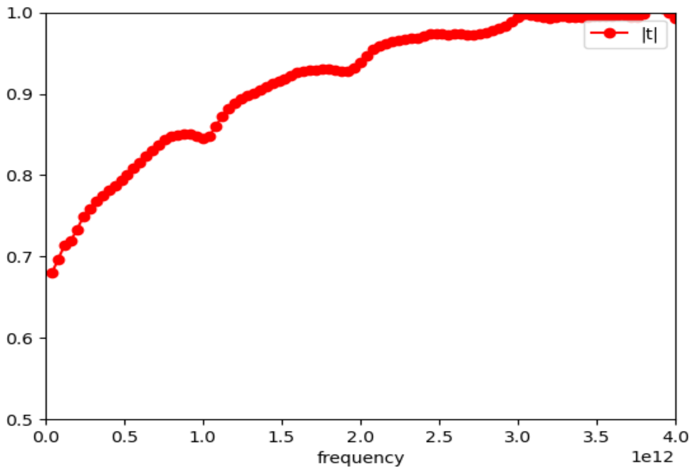


Figure.6: transmission spectrum of the pump-probe experiment. (f =ω/2**π**)

The 2-D Analytical Treatment

The results of the experiment can also be analytically solved by using a 2-D analytical treatment for the same problem. By the Huygen-Fresnel Law, every infinitesimal part of the electric field on a wavefront is a source of an outgoing spherical wave which, summed together, form the total transmitted wave. Therefore, in this 2-D scenario, the amplitude transmission can be calculated by a ratio of integrals shown in equation(3), and the scenario is shown in Figure.7.

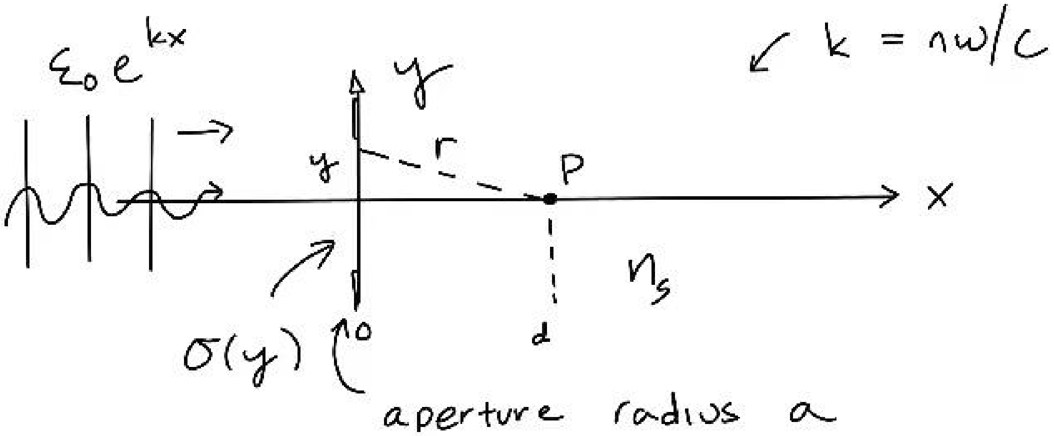


Figure.7: A diagram showing the 2-D analytical treatment. In this diagram, y is the distance from the center to a point on the sample; d is the distance from the center of the sample to the point of measurement; n_s is the refraction index of the dielectric medium attached to the sample

$$t(\omega) = \frac{\int_{-w/2}^{w/2} \frac{f(y) e^{in\omega/c\sqrt{y^2+d^2}}}{\sqrt{y^2+d^2}} dy}{\int_{-w/2}^{w/2} \frac{e^{in\omega/c\sqrt{y^2+d^2}}}{\sqrt{y^2+d^2}} dy} \tag{3}$$

(3)Here, n is the refraction index of the material, c is the speed of light, and other parameters have the same value as notated in Figure.7. f(y) can be described by equation (4).

$$f(y) = 1/(1 + \alpha\sigma(y)) \tag{4}$$

Where

$$\alpha = \mu_0 c/(1 + n_s) \tag{5}$$

And n_s is the substrate index of refraction, σ(y) is the function of y described by equation(2).

Using Mathematica, the transmission spectrum of the 2-D analytical treatment can be calculated and plotted out in the same graph as the simulation spectrum. In Figure.9, it is clear that the simulation result follows the analytical result pretty well with the same issue: discrepancies show up at lower and higher frequencies. But we can not simply expect the two results to completely agree, since the Huygen-Fresnel principle only approximates the solution using wave equations; the actual solutions are the solutions to Maxwell's equations. In general, the 2-D analytical treatment is a good case to compare to the simulation spectrum, and from the diagram, we can say that the simulation produces consistently accurate results.

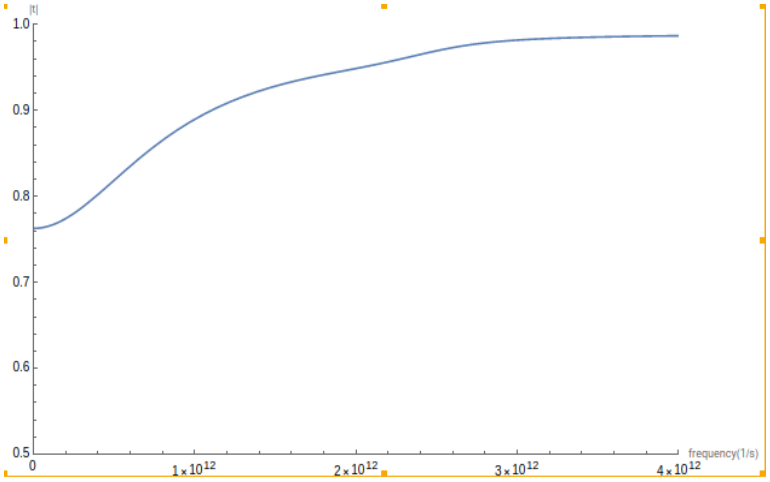


Figure.8: transmission spectrum calculated using Mathematica.

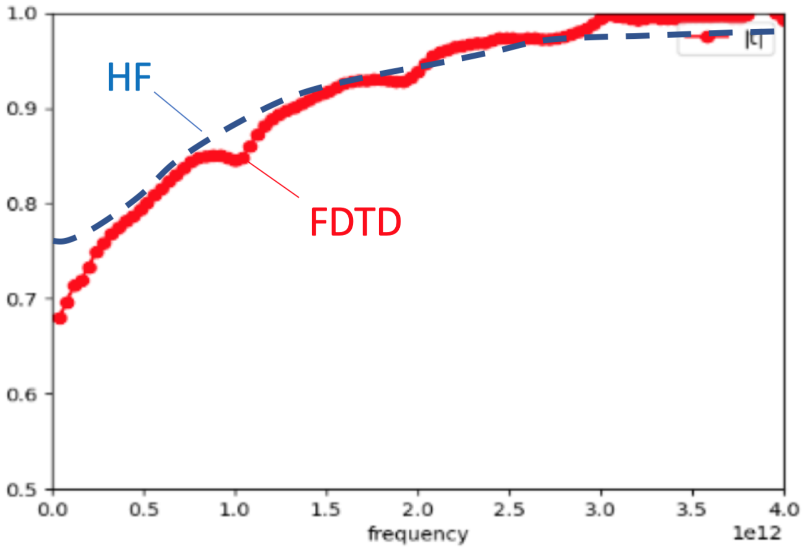


Figure.9: A diagram showing both the analytical result and the simulation result on the same scale.

Conclusion

In general, the simulation results are sufficiently correct The simulations were not perfect. There's plenty of room to improve the accuracy of the simulation results. However, a small increase in the resolution leads to a dramatic increase in the simulation time. Since the goal is to compare the experimental results with the simulation spectrum, accuracy requires a higher resolution for the task. But at what resolution can we get a more accurate result without costing too much time? At the current resolution, the simulation takes about 10 minutes to finish and produce results, when the resolution is doubled, the time it takes to complete the simulation becomes about 40 minutes. This is still an acceptable speed compared to actual lab experiment time. Therefore, FDTD simulation using "MEEP" has a relatively fast runtime, and it produces reasonably consistent and accurate results. My simulation results can be used to study the solar cell materials my research team is interested in, such as GaAsN and perovskite solar cells in the future. Here, I want to acknowledge Prof. James Heyman for providing me with advice and suggestions, my other research members: Erin Leary and Paige Stevenson for working with me in the labs and sharing their ideas, and Strategic Planning and Analysis Committee for recommending me for Collaborative Summer Research Grant from the Keck Foundation and Leonard Summer Research Funds.

References

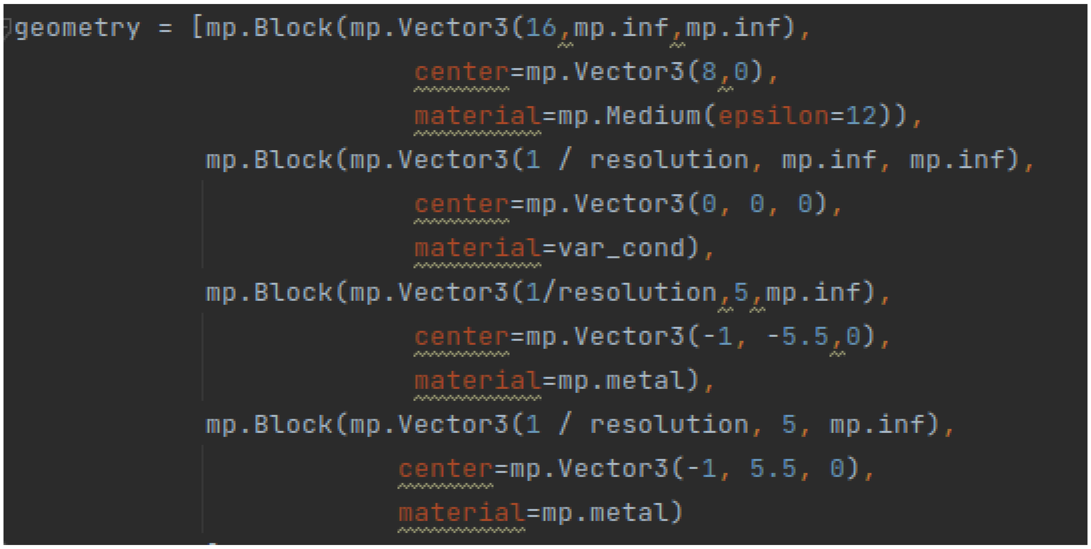
[1] J.E. Houle and D.M. Sullivan, *Electromagnetic Simulation Using the FDTD Method with Python*, 3rd ed. (Wiley, Newark, 2020).

[2] R.F. Pierret, *Semiconductor Device Fundamentals* (Pearson/Education, New Delhi (India), 2008).

[3] R.F. Pierret, *Advanced Semiconductor Fundamentals* (Pearson Education, Upper Saddle River, NJ, 2003).

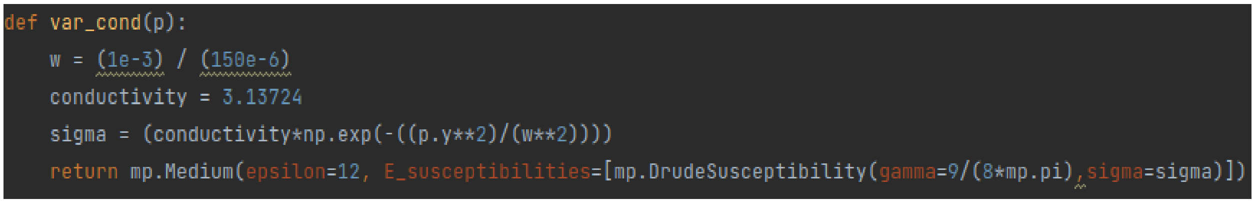
Appendix A

[1] Typical codes to define a dielectric profile in the simulation:



```
geometry = [mp.Block(mp.Vector3(16,mp.inf,mp.inf),
                         center=mp.Vector3(8,0),
                         material=mp.Medium(epsilon=12)),
            mp.Block(mp.Vector3(1 / resolution, mp.inf, mp.inf),
                         center=mp.Vector3(0, 0, 0),
                         material=var_cond),
            mp.Block(mp.Vector3(1/resolution,5,mp.inf),
                         center=mp.Vector3(-1, -5.5,0),
                         material=mp.metal),
            mp.Block(mp.Vector3(1 / resolution, 5, mp.inf),
                        center=mp.Vector3(-1, 5.5, 0),
                        material=mp.metal)
```

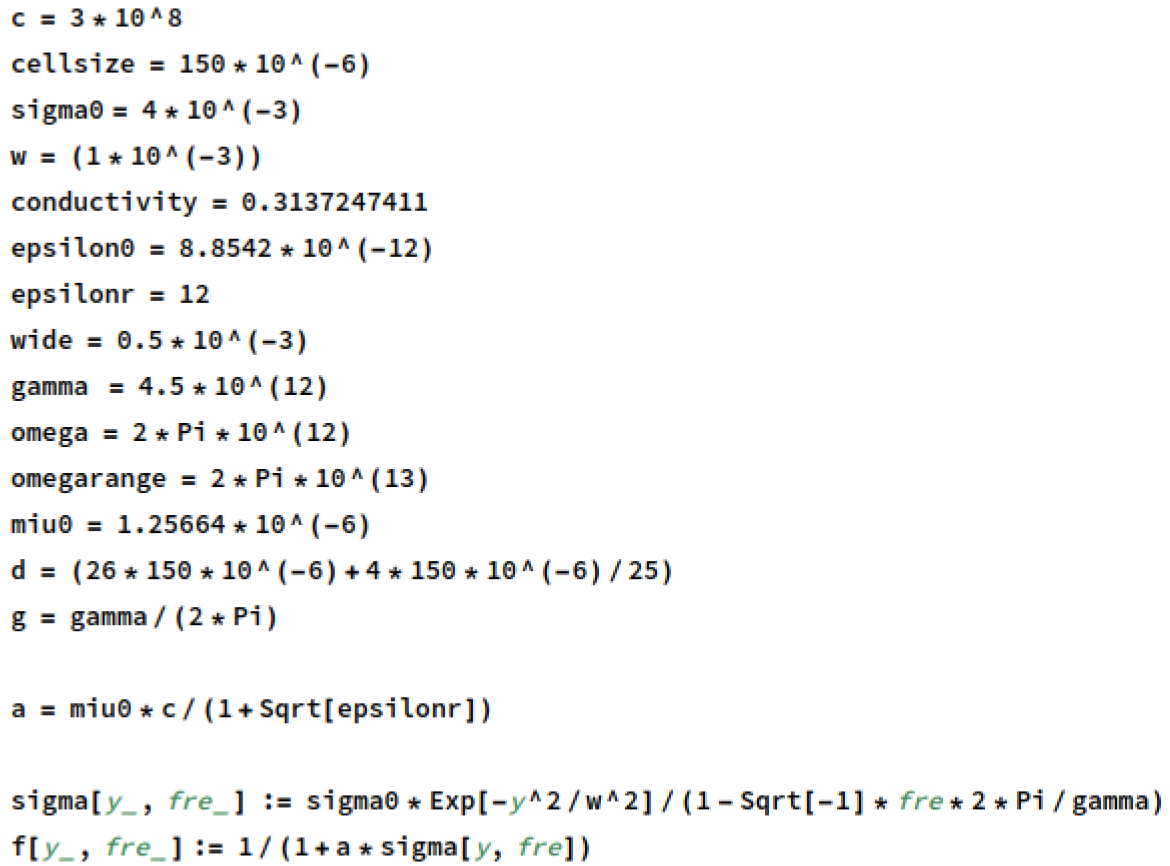
[2] Defining the position-varying conductivity:



```
def var_cond(p):
    w = (1e-3) / (150e-6)
    conductivity = 3.13724
    sigma = (conductivity*np.exp(-((p.y**2)/(w**2))))
    return mp.Medium(epsilon=12, E_susceptibilities=[mp.DrudeSusceptibility(gamma=9/(8*mp.pi),sigma=sigma)])
```

Appendix B

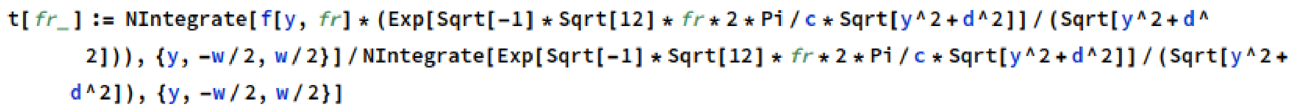
Mathematica Codes for calculating the 2-D analytical spectrum:



```
c = 3 * 10 ^ 8
cellsize = 150 * 10 ^ (-6)
sigma0 = 4 * 10 ^ (-3)
w = (1 * 10 ^ (-3))
conductivity = 0.3137247411
epsilon0 = 8.8542 * 10 ^ (-12)
epsilonr = 12
wide = 0.5 * 10 ^ (-3)
gamma  = 4.5 * 10 ^ (12)
omega = 2 * Pi * 10 ^ (12)
omegarange = 2 * Pi * 10 ^ (13)
miu0 = 1.25664 * 10 ^ (-6)
d = (26 * 150 * 10 ^ (-6) + 4 * 150 * 10 ^ (-6) / 25)
g = gamma / (2 * Pi)

a = miu0 * c / (1 + Sqrt[epsilonr])

sigma[y_, fre_] := sigma0 * Exp[-y ^ 2 / w ^ 2] / (1 - Sqrt[-1] * fre * 2 * Pi / gamma)
f[y_, fre_] := 1 / (1 + a * sigma[y, fre])
```



```
t[fr_] := NIntegrate[f[y, fr] * (Exp[Sqrt[-1] * Sqrt[12] * fr * 2 * Pi / c * Sqrt[y ^ 2 + d ^ 2]] / (Sqrt[y ^ 2 + d ^
     2])), {y, -w / 2, w / 2}] / NIntegrate[Exp[Sqrt[-1] * Sqrt[12] * fr * 2 * Pi / c * Sqrt[y ^ 2 + d ^ 2]] / (Sqrt[y ^ 2 +
    d ^ 2]), {y, -w / 2, w / 2}]
```