Transmission of Electromagnetic Waves Through Multiple Materials

Michael Daum
Department of Physics and Astronomy, Michigan State University, East Lansing, MI 48824
(Dated: April 24, 2024)

I. ABSTRACT

The transmission of electromagnetic waves through multiple boundaries is very important due to communications applications. While there's a package in python that is able to represent the transmission of a wave through one boundary, it does not allow for larger systems. The goal of this project was to design a program utilizing this python package that allows for more flexibility in amounts of boundaries in systems so that it could be further extended to real world applications. The implementation matched the single boundary scenario, however upon extending the number of boundaries the model exhibited major flaws in the amplitudes of the terms that were no longer in the first medium. The qualitative behavior however did match what was expected meaning that the amplitude calculation likely contains a flaw. Redesigning the backbone of the module as well as altering the methods used for calculation would be a next step in designing an accurate program. Additionally adding in features to model co-channel interference, calculate higher orders of transmission/reflection, modulate thickness, and having a materials dictionary with known values for common building materials would make this program feasible for being implemented in this sector.

II. INTRODUCTION

The propagation of electromagnetic waves (EM waves) through matter is very important to everyday life to many people. Many devices that are utilized by the average person use internet connectivity through a wireless connection to a Wi-Fi network or cell tower, which use EM waves. While the propagation of light is typically handled using the studies of optics with an index of refraction that can be determined through visible properties, EM waves must be handled differently due to their spectrum being outside of the visible range. Many material properties must be experimentally determined and when designing large scale Wi-Fi networks resources are often input into validating the effectiveness of designs that primarily take into account just the number of walls that the signal travels through. Due to the attenuation that's caused by this and additional interference from other signals, this can lead to designs needing to be reworked and more validations needing to be done. This requires a lot of manpower to be diverted to this process and time and effort needing to be reinvested to fix issues like these arising from hardware location, as well as costs due to potentially needing additional hardware for acceptable signal levels.

This makes it appealing to find a way to model wave propagation through these materials to determine levels of attenuation through each boundary before installation of hardware, allowing for more accurate initial designs and saving money, time, and effort for companies looking to employ this strategy. This paper investigates the implementation of this third boundary, the results of propagation through non-conductive and conductive materials, and then future work to make this project viable for the above solutions.

III. METHODOLOGY

In python, a package was released called em-waves [1], which had the capability to take material constants for two different materials, and animate the wave propagation through the boundary showing the changes that occur. However, this package only worked for one boundary, so it was chosen to serve as the basis for the model. This was done by using the .py file in which the code for the entire package was stored and modifying it to allow for multiple boundaries. The original package used a medium class, which stored relative values for permittivity (ε), permeability (μ), conductivity (σ), dielectric constant

$$\varepsilon_{eq} = \varepsilon_0 \varepsilon_r (1 + \frac{\sigma}{i\omega \varepsilon_0 \varepsilon_r})$$

[2], magnetic constant ($\mu_{eq} = \mu_0 \mu_r$) [3], and characteristic impedance

$$\zeta = \sqrt{\frac{\mu_{eq}}{\varepsilon_{eq}}}$$

[1]. This medium class had sufficient mutability for the purposes of this project so it was not changed.

The next class that the package contained was for a wave, and it contained a lot more attributes and methods. It took values for frequency and an amplitude function and then converted the frequency into an angular one. It then added vacuum as the default medium for all propagation. This then could be changed with the add mediums method which took 2 medium objects and then changed the attributes of the wave object to match the given mediums. It then had a method to calculate the wave number (k) of a medium, reflection coefficient:

$$\Gamma_{1,2} = \frac{\zeta_2 - \zeta_1}{\zeta_2 + \zeta_1}$$

transmission coefficient:

$$\tau_{1,2} = \frac{2\zeta_2}{\zeta_2 + \zeta_1}$$

the skin depth of a medium, velocity in a medium, wavelength in a medium, and power density transfer:

$$\frac{|A|^2}{2|\zeta_1|}(1 - |\Gamma_{1,2}|)$$

where the amplitude is represented by A, and a print function which showed the values of all quantities. Next was a show function which contained the animation and calculations for the wave propagation with a time, amplitude, and y limits. It used lambda functions to take values for position in space and time, then plotted returned amplitude based off of the value of the amplitude function at that point in spacetime with the value of k for that medium. Once this was done it was a simple case of animating the data that was had by setting the data for each line that was plotted in matplotlib. Finally the subclass of Sine was used which just added an amplitude function for a sine wave using k, z, and t as well as defined default y limits and time values for this function.

The changes that were made started with the way mediums were instantiated in the wave object. Mediums were stored as an array rather than their own attributes, with mediums now being used in the construction of the object with a default of none. If there were no mediums vacuum would be set for 2 mediums, otherwise if there was only 1 given the medium would be appended to the list after vacuum, otherwise if a list was given it would iterate through adding mediums. The add mediums function was changed to accommodate this structure with a remove mediums method being created as well. k was only dependent on a single medium allowing it to stay the same, however the Γ calculation method changed to take a list of mediums and returned a list of the reflection coefficients between them. A similar structure was followed for the τ calculation, skin depth, and power density transmission functions.

The most substantial changes however occurred in the show function, which needed to now accommodate all of these changes the x limits needed to be calculated based on number of mediums, with points in space chosen such that they were stored in a list based off of number of mediums. k and wavelength (λ) were calculated by iterating through mediums and the lists for coefficents of reflection and transmission were stored as well. Empty lists for functions for each medium were instantiated and each medium was looped through to create functions for them, with a very similar process followed for plotting and animated, being adjusted for having multiple waves. Additionally vertical lines were added at boundaries so that it was clear where boundaries were occurring. Changes were also made to account for labeling on the plots in an iterative fashion so that any number of materials could be represented clearly with minimal user changes.

The code was then tested by trying the same parameters for the original em-waves package as well as the modified package allowing for multiple boundaries. These results are shown in Fig. 1 and Fig. 2.

IV. RESULTS AND DISCUSSION

Fig num	ε_r	μ_r	$sigma$
3	2	1	0
5	5	3	.81

TABLE I. Medium 2 material constants for modified code

Now that the code has been verified to one boundary, the model was extended for multiple boundaries, however this is where the limitations of the code start to show. The values for each parameter are shown in table 1. As seen in

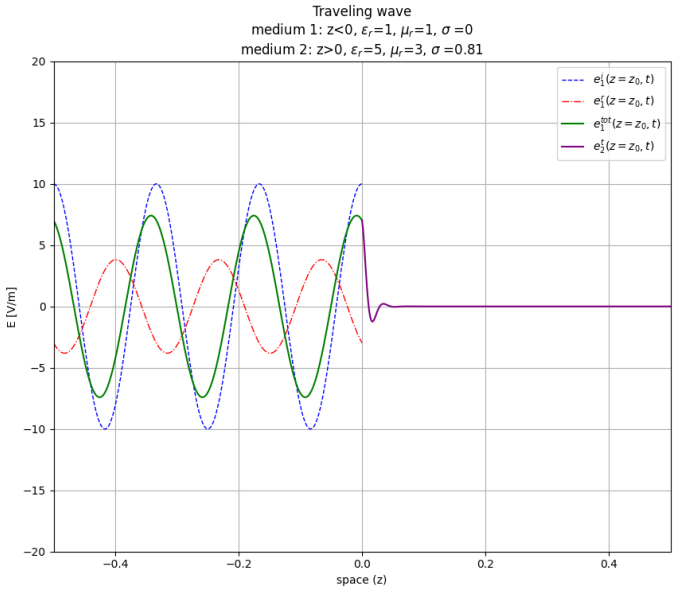

FIG. 1. em-waves result for vacuum and $\varepsilon_r = 5$, $\mu_r = 3$, and $\sigma = .81$

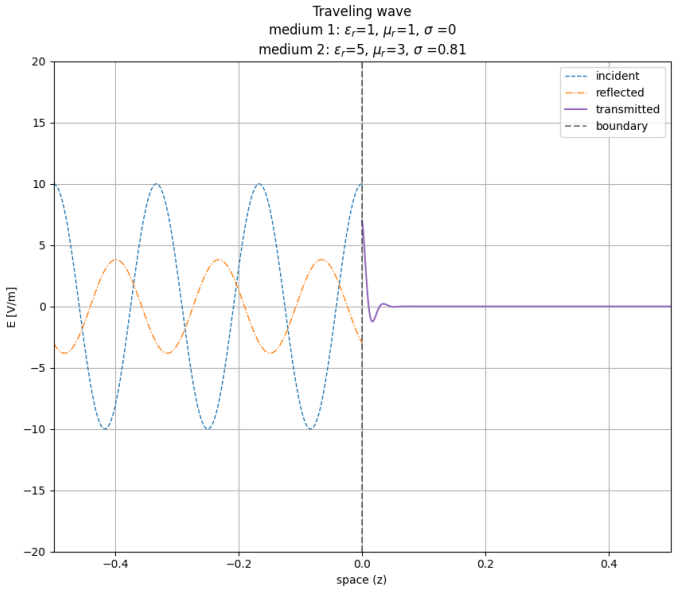

FIG. 2. Modified package result for vacuum and $\varepsilon_r = 5$, $\mu_r = 3$, and $\sigma = .81$

Fig. 3, unexpected results occurred when the third boundary condition is added, where amplitudes are far outside of expected values. There will be more about these failings in the conclusions, however while the amplitudes are wrong, it can still be seen that the behavior that they follow is still expected, making the results valid qualitatively. The issues with the amplitudes can be seen in Fig. 3, where it's clear that the amplitude in medium 2 is larger, meaning that the wave has gained intensity passing through this material, and then retains this intensity through material 3 which is free space once more.

As can be seen in Fig. 4, the amplitude of the transmitted wave should in fact decrease rather than increase. Due to constructive interference of the reflected wave in medium 2 this could cause a higher amplitude in Fig. 3 close to magnitude in that which is seen, however this is uncertain and the amplitude is only supposed to be that from the transmitted wave from medium 1 to 2. Additionally, this interference disappears in medium 3 and should return to the initial amplitude if no energy is lost, however this is not the case and instead the wave remains at this higher amplitude. Additionally it appears that the wavelength is shorter entering this last medium than the first medium,

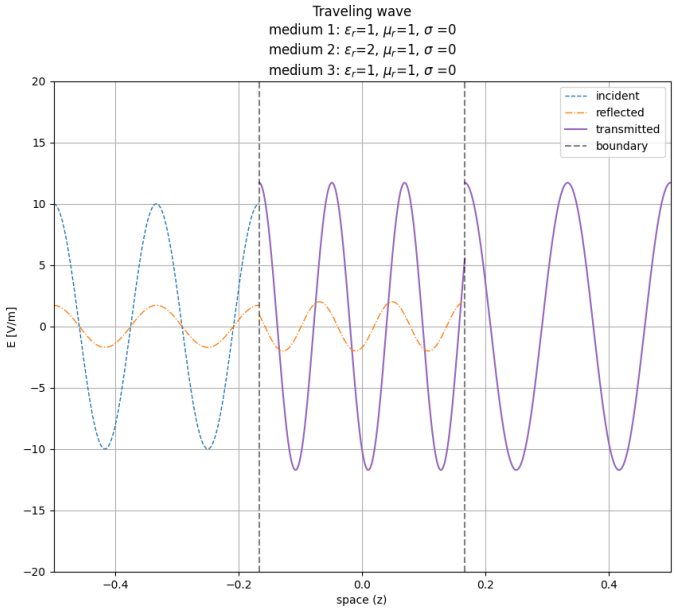

FIG. 3. Modified package result for $\varepsilon_r = 2$, $\mu_r = 1$, and $\sigma = 0$ surrounded by vacuum on each side

making the results once again only usable on a qualitative level.

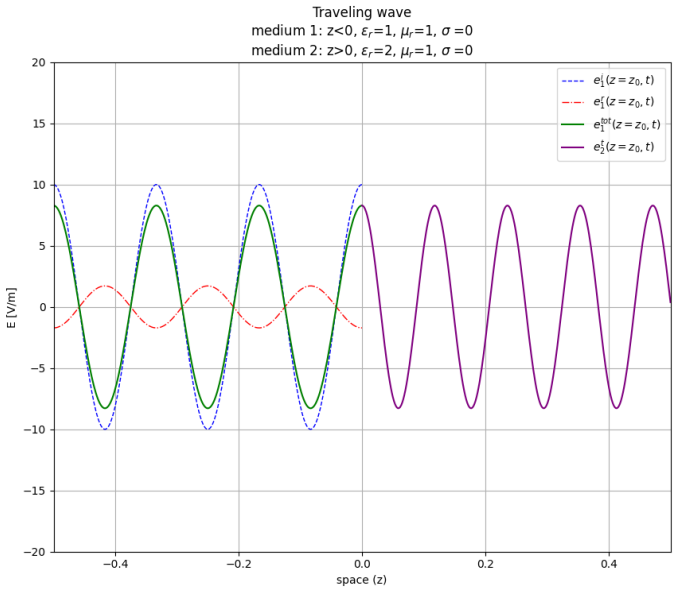

FIG. 4. em-waves result for $\varepsilon_r = 2$, $\mu_r = 1$, and $\sigma = 0$ with incident beam in vacuum

Figure 5 shows the same medium 2 that was used in Fig. 1 and Fig. 2, showing that once again adding a third medium causes massive issues, with the transmitted wave in medium 2 being higher amplitude than that of the incident wave in medium 1. Subsequently, there's strange behavior in the reflected wave where it increases in amplitude after reflection. Nevertheless this is the trend the amplitude should exhibit when reflecting, decreasing as it continues to travel through the medium, but it is far too large in amplitude compared to what it should be. Additionally, it can again be seen in medium 3 that the amplitude grows and matches what is displayed in medium 2, but unlike Fig. 3, the wavelength appears to be near the expected value.

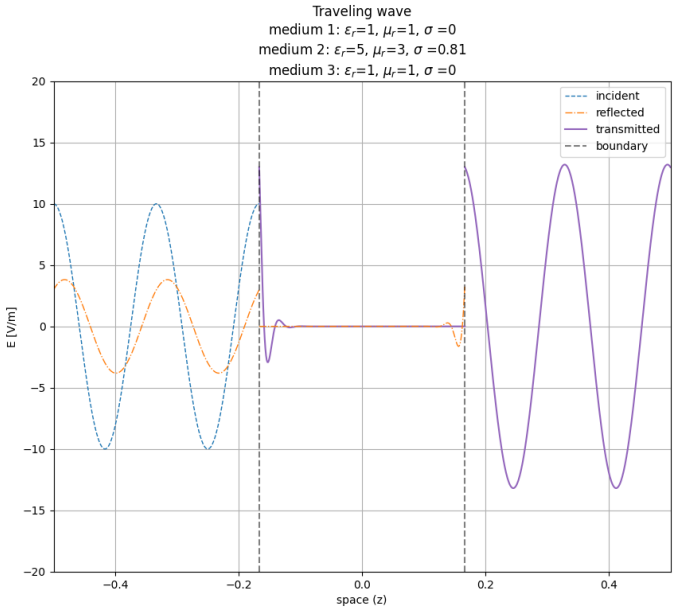

FIG. 5. Modified package result for $\varepsilon_r = 5$, $\mu_r = 3$, and $\sigma = .81$ surrounded by vacuum on each side

V. CONCLUSIONS AND PERSPECTIVES

The results when adding a third boundary indicate a flaw in the methods used to calculate the amplitudes of transmitted waves and of reflected waves following the first. It is expected that part of the issue is directly related to how the code represents space and calculated the coefficients based on the idea of a boundary being placed at the zero point in space in the original python package. Although transformations were done to try and circumvent this, they seem to have failed and various attempts of changing other parameters while still maintaining the skeleton of the original package failed. The belief of the author is that the package's design was never meant to be extended to this application and that it would be easier to either completely redesign the package only keeping the medium class and redesigning much of the wave class. Another potential solution would be to export the values from the second medium in the working code and then import those into a new wave object somehow, which again would likely require an entire redesign of the program.

Ultimately this program in its current state is only useful as a qualitative introduction and it also doesn't take interference from higher order transmissions/reflections which would also make the code more accurate. Furthermore adjusting the thickness of materials is another very important factor to consider and is not available in the current state of the program, making it not usable for the many of the purposes mentioned in the introduction. It would likely be better to use the power density functions to calculate transmissions and rewrite the reflections in terms of the transmitted wave and incident wave. The higher order transmissions and reflections are significantly smaller and had less impact, so approximating them as zero was most likely fine for even fairly technical applications. In addition, creating a program that combines the different mediums through calculating sets of 2 mediums at once using the previous medium's result and then plotting it all together is likely a better solution that trying to combine all the calculations into a single process.

As previously stated, future improvements could be resolving the issues with amplitudes and adding in variable thickness. Other work could include adding in default materials that are commonly used in buildings, writing code to save the animations/view them in jupyter notebooks rather than through command line interface, and adding in a model for co-channel interference (waves already populating the area causing destructive interference with the waves being studied).

VI. REFERENCES

[1] - Francesco Urbani, em-waves, Github, 2020

[2] - Jordon, Edward C., Balman, Keith G. Electromagnetic Waves and Radiating Systems, Prentice-Hall, New Jersey, 1968, p. 178

[3] - David J. Griffiths, Introduction to Electrodynamics, Cambridge University Press, Cambridge, 2017, p. 401-424

VII. APPENDICES

Useful equations:

- Sine wave: $E = E_0 e^{i(\mathbf{k}\cdot\mathbf{r}-\omega t)}$
- Wave number: $k = \omega\sqrt{\mu_{eq}\varepsilon_{eq}}$
- Velocity: $v = \frac{1}{\sqrt{\mu_{eq}\varepsilon_{eq}}}$
- Wavelength: $\lambda = \frac{v}{f}$
- Angular frequency: $\omega = 2\pi f$