Visualizing Electric Dipole Radiation

Romas Jatulis

March 20, 2024

1 Project Overview

The goal of this project was to visualize dipole radiation. I also wanted to focus on the physical applications of dipole radiation with antenna design. I did this by incorporating physical adjustable values into the dipole: frequency, dipole moment, and region of interest. I also wanted the result to show meaningful visuals on the scale of meters. This is a distance that we can easily imagine and it gives a good picture of what radiation might actually look like in the room around us.

The first and most challenging task for me was deciding how to visualize this radiation. There are a number of different values and fields that describe radiation. I chose to focus on the Poynting Vector which is the cross product of the Electric and Magnetic fields. The Poynting Vector is significant because its average is intensity. Visualizing the Poynting Vector gives a good idea of where the radiation is going and how strong it is.

Next was to find the Poynting Vector for a dipole. This required brushing up on my electromagnetism and researching the proper equations and approximations to describe the field. Since I wanted to focus on antenna design and large distances I took advantage of the far field approximation. I didn't want to use equations that were oversimplified and basic. I also didn't want something too complex and general. After significant research and testing, I ended up using the following field equations found in Jackson electromagnetism:

$$E = \frac{1}{4\pi\epsilon_0}\left\{\frac{\omega^2}{c^2 r}(\hat{r}\times\mathbf{p})\times\hat{r} + \left(\frac{1}{r^3} - \frac{i\omega}{cr^2}\right)[3\hat{r}(\hat{r}\cdot\mathbf{p}) - \mathbf{p}]\right\} e^{i\omega r/c} e^{-i\omega t} \tag{1}$$

$$B = \frac{\omega^2}{4\pi\epsilon_0 c^3}(\hat{r}\times\mathbf{p})\left(1 - \frac{c}{i\omega r}\right)\frac{e^{i\omega r/c}}{r} e^{-i\omega t} \tag{2}$$

For my visuals, I assumed that the dipole moment p is in the positive $\hat{z}$ direction. Putting these in Python and finding the cross-product to get the Poynting vector made the program very slow. This is a ton of calculations to make for each frame of an animation. Instead, I used the Poynting Vector components:

$$S_r = \frac{cp^2k^4}{8\pi r^2}\sin^2\theta\left[\left(1 - \frac{2}{(kr)^2}\right)\cos(2\Phi) - \left(\frac{2}{kr} - \frac{1}{(kr)^3}\right)\sin(2\Phi)\right] \tag{3}$$

$$S_\theta = \frac{cp^2k^3}{4\pi r^3}\sin\theta\cos\theta\left[\left(1 - \frac{1}{(kr)^2}\right)\sin(2\Phi) + \frac{2}{kr}\cos(2\Phi)\right] \tag{4}$$

with $\Phi = k\cdot r - \omega t$. These equations made my code much more simple and fast.

For my program, I created three different visualizations of dipole radiation. The first was the intensity profile for a radiating dipole. This gives a 3D view of what the radiation looks like in space. It is hard to visualize the specific magnitude and direction for such a complex behavior on one small chart, so I added visualizations to see what is happening inside the 3D plot.

Next, I made two animations of the dipole radiation in the x-y and x-z planes. Because of symmetry, this allows us to see how the dipole radiation behaves everywhere. These animations are contour plots of the Poynting Vector in space. The magnitude of the Poynting Vector is visualized by the color and the animation shows the temporal dependence of the Poytning Vector. Combining these creates a really good visualization that shows the intensity of the radiation and where the radiation is traveling as time evolves. The alternative to this route would have been arrows with the length representing magnitude. This was much more difficult to see.

The theory and math ended up being much more difficult than the coding. Working a lot on the methods and equations for visualization simplified the program greatly. Having the simple program meant I could put all the computer power toward the visualization instead of having to do a bunch of calculations. The biggest improvement to my code's performance was definitely directly using the Poynting Vector components.

The main complexion in my code came from creating the animation. This required a lot of fine-tuning of parameters and scales to make the smoothest animation. I made comments in my code about my thought process behind these adjustments.

2 Results

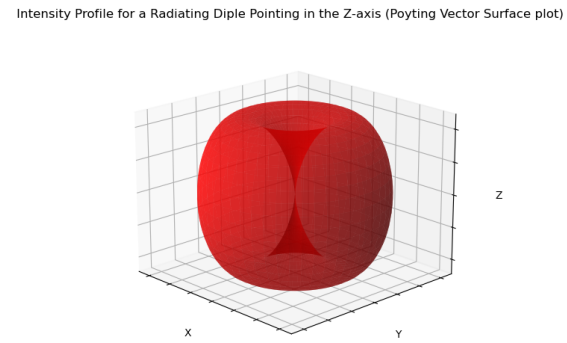

Figure 1: *Intensity profile for a radiating dipole oriented in the z-axis. This is a surface plot of the Poynting Vector. I wanted to include this to get an idea of what the radiation looks like in 3D, but the main part of the program is the animations.*

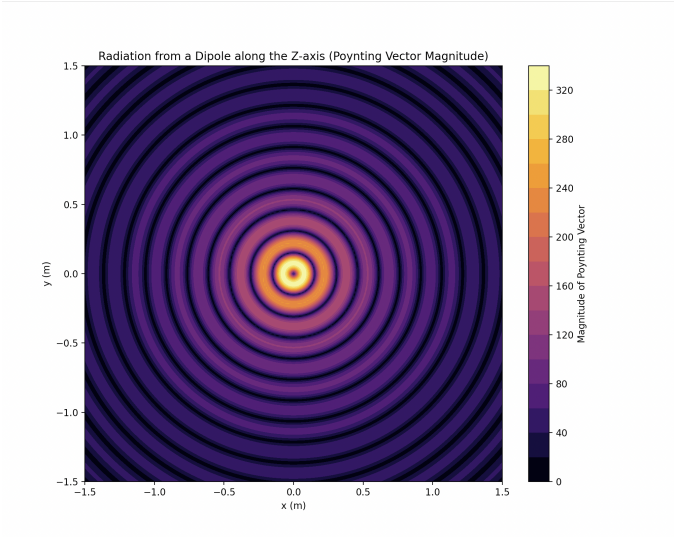

Figure 2: *This is the contour plot of the Poynting Vector magnitude in the xy plane. The dipole is in the z axis, oriented out of the page. The figure is animated when the program is ran.*

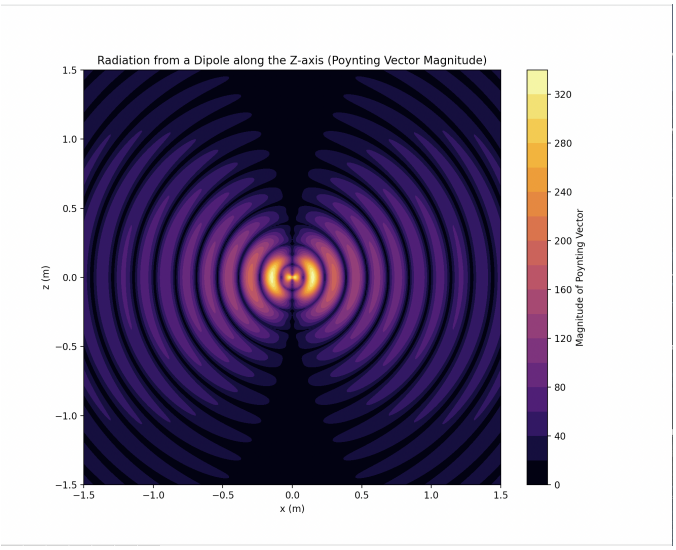

Figure 3: *This is the countour plot of the Poynting Vector magnitude in the xz plane. This dipole is at the origin oriented upward along the z axis. The figure is animated when the program is ran.*

3 Conclusion

I am very happy with how the animations and visualizations turned out, but I would like to make some improvements and additions. In my outline, I said I wanted to focus on antenna design and I wanted to see the radiation with multiple dipoles. I thought that this would be a simple solution of the superposition of two of these plots, but the problem would require more attention. For dipoles close together, the solution would be complicated and calculation intensive as the near field would also have to be considered.

This project was interesting for me. I learned about the many different types and shapes of antennae as well as the role of a dish or reflector. The theory was more intricate than I predicted, but I would love to see what I can create with different parameters, systems, and methods of modeling.

4 Notes/Instructions

The program is pretty simple and intuitive. Choose a plot to view and when you close the plot you can choose a different plot to view. The parameters can be changed at the top of the program, but you have to be careful. For example, the equations with the far field approximation break down if the region of interest is too small or the wavelength is too large.