EE2703 : Applied Programming Lab
End-Semester Examination

Andapally Snehitha
EE20B008

May 12, 2022

Introduction

The currents in a half wave dipole antenna are assumed to be as

$$I(z) = \begin{cases} I_m * sin(k(l - z)), & 0 \leq z \leq l \\ I_m * sin(k(l + z)), & -l \leq z < 0 \end{cases} \tag{1}$$

We are tasked with fnding the validity of this assumption. We will do this

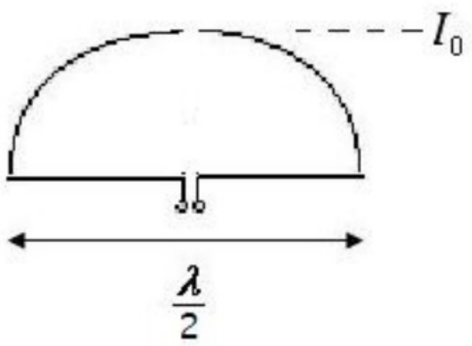

Figure 1: Half wave dipole antenna

by finding the magnetic field using two methods:

Ampere's law:

From Ampere's law, we get:

$$2\pi a H_\phi(z_i) = J_i$$

Vector potential:

We know that the curl of the vector potential is the magnetic field. So, we will find the vector potential and take it's curl.
Vector potential along the circumference by currents at different parts of the antenna is given by

$$\vec{A}(r, z) = \frac{\mu_0}{4\pi} \int_z \frac{I(z')\hat{z}e^{-jkR}dz'_j}{R}$$

Taking the curl of this equation will relate H to J. We now have two equations for H and J. So we can solve for J from the equations.

Problem 1

1. Create z vector of zeros of length 2N+1. 2. Create u vector of zeros of length 2N-2. 3. Create I vector of zeros of length 2N+1. 4. Create J vector of zeros of length 2N-2.

```
    z = np.zeros(2*N+1)
    u = np.zeros(2*N-2)
    I = np.zeros(2*N+1)
    J = np.zeros(2*N-2)
```

Problem 2

Create an identity matrix of size 2N-1 and divide each value by 2*pi*a.

```
    M = np.identity(2*N-2, dtype = float)
    M = M/(2*np.pi*a)
```

The M matrix is

$$\begin{pmatrix} 15.91549431 & 0 & 0 & 0 & 0 & 0 \\ 0 & 15.91549431 & 0 & 0 & 0 & 0 \\ 0 & 0 & 15.91549431 & 0 & 0 & 0 \\ 0 & 0 & 0 & 15.91549431 & 0 & 0 \\ 0 & 0 & 0 & 0 & 15.91549431 & 0 \\ 0 & 0 & 0 & 0 & 0 & 15.91549431 \end{pmatrix}$$

Problem 3

1. Create meshgrid by taking inputs as z, u.
2. Fill Rz
3. Create another meshgrid by taking inputs as u, u.
4. Fill Ru
5. Create Pb
6. Create Pij

The Rz matrix is

$$\begin{pmatrix}
0.13 & 0.01 & 0.13 & 0.25 & 0.38 & 0.5 & 0.63 & 0.75 & 0.88 \\
0.25 & 0.13 & 0.01 & 0.13 & 0.25 & 0.38 & 0.5 & 0.63 & 0.75 \\
0.38 & 0.25 & 0.13 & 0.01 & 0.13 & 0.25 & 0.38 & 0.5 & 0.63 \\
0.63 & 0.5 & 0.38 & 0.25 & 0.13 & 0.01 & 0.13 & 0.25 & 0.38 \\
0.75 & 0.63 & 0.5 & 0.38 & 0.25 & 0.13 & 0.01 & 0.13 & 0.25 \\
0.88 & 0.75 & 0.63 & 0.5 & 0.38 & 0.25 & 0.13 & 0.01 & 0.13
\end{pmatrix}$$

The P matrix is

$$\begin{pmatrix}
(124.943.93j) & (9.23.83j) & (3.533.53j) & (02.5j) & (0.771.85j) & (1.181.18j) \\
(9.23.83j) & (124.943.93j) & (9.23.83j) & (1.273.08j) & (02.5j) & (0.771.85j) \\
(3.533.53j) & (9.23.83j) & (124.943.93j) & (3.533.53j) & (1.273.08j) & (02.5j) \\
(02.5j) & (1.273.08j) & (3.533.53j) & (124.943.93j) & (9.23.83j) & (3.533.53j) \\
(0.771.85j) & (02.5j) & (1.273.08j) & (9.23.83j) & (124.943.93j) & (9.23.83j) \\
(1.181.18j) & (0.771.85j) & (02.5j) & (3.533.53j) & (9.23.83j) & (124.943.93j)
\end{pmatrix}$$

The Pb matrix is

$$\begin{pmatrix} ((1.273.08j) & (3.533.53j) & (9.23.83j) & (9.23.83j) & (3.533.53j) & (1.273.08j)) \end{pmatrix}$$

Problem 4

1. Compute Qij
2. Compute Qb

The Q matrix is

$$\begin{pmatrix}
(99.5210.001j) & (0.0540.001j) & (0.0080.001j) & (0.0010.001j) & (0.0010.001j) & 0.001j \\
(0.0540.001j) & (99.5210.001j) & (0.0540.001j) & (0.0030.001j) & (0.0010.001j) & (0.0010.001j \\
(0.0080.001j) & (0.0540.001j) & (99.5210.001j) & (0.0080.001j) & (0.0030.001j) & (0.0010.001j \\
(0.0010.001j) & (0.0030.001j) & (0.0080.001j) & (99.5210.001j) & (0.0540.001j) & (0.0080.001j \\
(0.0010.001j) & (0.0010.001j) & (0.0030.001j) & (0.0540.001j) & (99.5210.001j) & (0.0540.001j \\
0.001j & (0.0010.001j) & (0.0010.001j) & (0.0080.001j) & (0.0540.001j) & (99.5210.001
\end{pmatrix}$$

The Qb matrix is

$$\begin{pmatrix} (0.0030.001j) & (0.0080.001j) & (0.0540.001j) & (0.0540.001j) & (0.0080.001j) & (0.0030.001j) \end{pmatrix}$$

Problem 4

1. Compute J
2. Get I vector by adding currents at z = 0, ±l to J
3. Plot the computed value and the assumed value of I.

```

Y = Im*Qb
    X = M-Q
    X1= np.linalg.inv(X)
    J = np.matmul(X1,Y)
    #Inserting the boundary conditions
    I = np.insert(J,0,0)
    I = np.append(I,0)
    I = np.insert(I,N,complex(Im))
```

Let us plot the currents for N = 4: Let us plot the currents for N = 100:

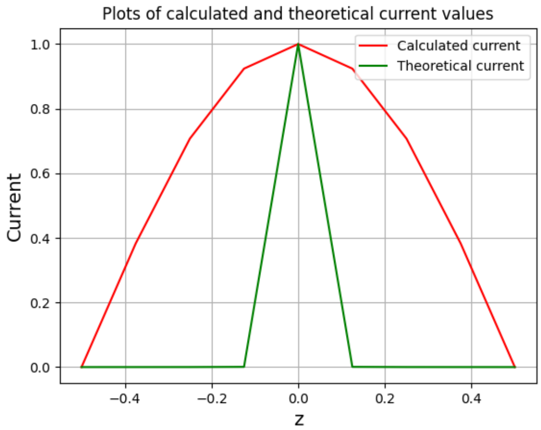

Figure 2: Current vs z (N = 4)

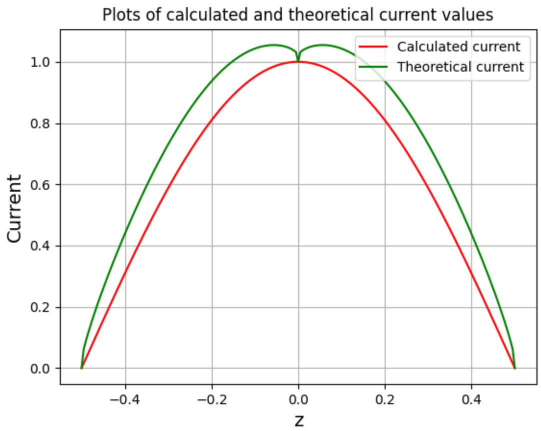

Figure 3: Current vs z (N = 100)

We can see that N = 100 is a much better approximation than N = 4. The reason the two plots do not match perfectly is because the assumption that the currents in the antenna vary sinusoidally is not perfectly valid. The true nature of the currents will be revealed when we compute J with a very high number of points.

Conclusion

In this problem, we solved the dipole current by finding the magnetic field around the antenna using two different methods, Ampere's law and the vector potential method. It turns out that the current we get is not completely sinusoidal. This suggests that the sinusoidal approximation is not completely valid. Nevertheless, the sinusoidal approximation is sufficient for real-world applications.