EE2703 : Applied Programming Lab

End Semester Exam

Antenna Currents in a Half-Wave Dipole Antenna

Chagari Koushal Kumar Reddy
EE20B023

May 13, 2022

Contents

1 Problem Statement

There is a long wire carrying current $I(z)$ in a dipole antenna of half length 50cm. The standard analysis of dipole antennas assumes that the current distribution is given by:

$$I(z) = \begin{cases} Im\sin(k(l-z)) & 0 \le z \le l \\ Im\sin(k(l+z)) & -l \le z < 0 \end{cases}$$

This assumed value of current is generally used to find the radiation field. We want to know whether this was a good assumption by calculating the actual values of the current in the dipole. We finally plot these 2 graphs in the same plot and compare them.

2 Pseudocodes, Codes and Plots

2.1 Section 1

2.1.1 Pseudocode

1. INIT All required constants (c,μ_o, l, dz, λ, k)
2. INIT Location of known, unknown currents (z, u)
3. INIT Vector contaning all currents, unknown currents (I, J)
4. INIT Boundary conditions of I ($I[0] = I[2N] = 0$, $I[N] = I_m$)

2.1.2 Code

```
i = p.linspace(-N,N,2*N+1)
z = i*dz
I = p.zeros((2*N+1),dtype = complex)
I[0] = 0 ; I [2*N] = 0 ; I[N] = Im

# 2N - 2 locations of unknown currents
# Deleting the first , middle and last entries
u = p.delete(z,(0,N,2*N))
J = p.zeros(2*N-2)
```

2.1.3 Explanation

The locations at which we compute the currents are given by z. z has $2N+1$ elements varying from $-l$ to l. The $2N+1$ currents are stored in the I vector. Since we know the values of I

at $0, N, 2N$ there are $2N-2$ unknown currents. The corresponding unknown current densities are stored in J. The corresponding unknown current locations are also computed and stored in the vector u.

2.1.4 Output Vectors (For $N = 4$)

- $z : \begin{bmatrix} -0.5 & -0.375 & -0.25 & -0.125 & 0. & 0.125 & 0.25 & 0.375 & 0.5 \end{bmatrix}$
- $u : \begin{bmatrix} -0.375 & -0.25 & -0.125 & 0.125 & 0.25 & 0.375 \end{bmatrix}$
- $I : \begin{bmatrix} 0. & 0. & 0. & 0. & 1. & 0. & 0. & 0. & 0. \end{bmatrix}$
- $J : \begin{bmatrix} 0. & 0. & 0. & 0. & 0. & 0. \end{bmatrix}$

2.2 Section 2

2.2.1 Pseudocode

1. FUNCTION: M
2. INPUT: Number of sections in each half of antenna N
3. CALCULATE Matrix M using identity matrix and radius of wire a
4. RETURN Matrix M
5. ENDFUNCTION

2.2.2 Code

```
def M(n):
# Matrix M of order 2*N-2 by 2*N-2
    return (1/(2*pi*a))*p.identity(2*n-2)
```

2.2.3 Explanation

In the first method to calculate $H_\phi(z, r = a)$, as explained above a matrix M needs to be created. This function generates such a matrix M taking input as N. Here a is the radius of the wire.

$$M = \frac{1}{2\pi a} \begin{pmatrix} 1 & \dots & 0 & 0 & \dots & 0 \\ \multicolumn{6}{c}{\dots\dots\dots\dots\dots\dots} \\ 0 & \dots & 1 & 0 & \dots & 0 \\ 0 & \dots & 0 & 1 & \dots & 0 \\ \multicolumn{6}{c}{\dots\dots\dots\dots\dots\dots} \\ 0 & \dots & 0 & 0 & \dots & 1 \end{pmatrix}$$

2.2.4 Output Vectors (For $N = 4$)

- $M = \begin{bmatrix} 15.92 & 0 & 0 & 0 & 0 & 0 \\ 0 & 15.92 & 0 & 0 & 0 & 0 \\ 0 & 0 & 15.92 & 0 & 0 & 0 \\ 0 & 0 & 0 & 15.92 & 0 & 0 \\ 0 & 0 & 0 & 0 & 15.92 & 0 \\ 0 & 0 & 0 & 0 & 0 & 15.92 \end{bmatrix}$

2.3 Section 3

2.3.1 Pseudocode

1. CALCULATE Distances for all pairs of z_i, z_j for i, j from 0 to $2N$ (R_z)
2. CALCULATE Distances for unknown currents by removing $z = 0, N, 2N$ (R_u)
3. CALCULATE Contribution to vector potential by unknown current sources (P)
4. CALCULATE Contribution to vector potential by current I_N (P_B)

2.3.2 Code

```
# Meshgrid generates 2D matrices which are used to find the
# distance between 2 points zi, zj as i,j go from 0 to 2N+1

zj,zi = p.meshgrid(z,z)
zij = p.zeros((2*N+1,2*N+1))
zij = (zi - zj)**2
Rz = p.zeros((2*N+1,2*N+1))
# Since radius of the wire, a = r
Rz = p.sqrt(a**2 + zij)
print("Rz : ", (Rz).round(2))

# Here we are calculating distances for unknown current locations

zj,zi = p.meshgrid(u,u)
zij = p.zeros((2*N-2,2*N-2))
zij = (zi - zj)**2
Ru = p.zeros((2*N-2,2*N-2))
# Since radius of the wire, a = r
Ru = p.sqrt(a**2 + zij)
print("Ru : ", (Ru).round(2))

# Calculation of matrices P and Pb

j = complex(0,1)              # Defining j
```

```
P = p.zeros((2*N-2,2*N-2), dtype = complex)
P = mu0*p.exp(-j*k*Ru)*dz/(4*pi*Ru)
print("P*1e8 : ", (P*1e8).round(2))

Pb = p.zeros((2*N-2),dtype = complex)
# From Rz[:N], we don't need distances to z = 0, z = N, z = 2N
Rz_del = p.delete(Rz[:,N],(0,N,2*N))
Pb = mu0*p.exp(-j*k*Rz_del)*dz/(4*pi*Rz_del)
print("Pb*1e8 : ", (Pb*1e8).round(2))
```

2.3.3 Explanation

R_{ij} is the distance of the point $\vec{r} + z_i\hat{z}$ and the source $z_j\hat{z}$. So R_{ij} value is calculated as:

$$R_{ij} = a^2 + (z_i - z_j)^2$$

Another name for the above is R_z. Similarly we create R_u which is a vector of distances of unknown currents. We also compute the matrix P which is given by:

$$P_{ij} = \frac{\mu_o}{4\pi}\frac{exp(-jkR_{ij})}{R_{ij}}dz$$

P_B is the vector corresponding to the contribution to the vector potential due to current I_N and is given by:

$$P_B = \frac{\mu_o}{4\pi}\frac{exp(-jkR_{iN})}{R_{iN}}dz$$

2.3.4 Output Vectors (For $N = 4$)

- R_z : $\begin{bmatrix} 0.01 & 0.13 & 0.25 & 0.38 & 0.5 & 0.63 & 0.75 & 0.88 & 1 \\ 0.13 & 0.01 & 0.13 & 0.25 & 0.38 & 0.5 & 0.63 & 0.75 & 0.88 \\ 0.25 & 0.13 & 0.01 & 0.13 & 0.25 & 0.38 & 0.5 & 0.63 & 0.75 \\ 0.38 & 0.25 & 0.13 & 0.01 & 0.13 & 0.25 & 0.38 & 0.5 & 0.63 \\ 0.5 & 0.38 & 0.25 & 0.13 & 0.01 & 0.13 & 0.25 & 0.38 & 0.5 \\ 0.63 & 0.5 & 0.38 & 0.25 & 0.13 & 0.01 & 0.13 & 0.25 & 0.38 \\ 0.75 & 0.63 & 0.5 & 0.38 & 0.25 & 0.13 & 0.01 & 0.13 & 0.25 \\ 0.88 & 0.75 & 0.63 & 0.5 & 0.38 & 0.25 & 0.13 & 0.01 & 0.13 \\ 1 & 0.88 & 0.75 & 0.63 & 0.5 & 0.38 & 0.25 & 0.13 & 0.01 \end{bmatrix}$

- R_u : $\begin{bmatrix} 0.01 & 0.13 & 0.25 & 0.5 & 0.63 & 0.75 \\ 0.13 & 0.01 & 0.13 & 0.38 & 0.5 & 0.63 \\ 0.25 & 0.13 & 0.01 & 0.25 & 0.38 & 0.5 \\ 0.5 & 0.38 & 0.25 & 0.01 & 0.13 & 0.25 \\ 0.63 & 0.5 & 0.38 & 0.13 & 0.01 & 0.13 \\ 0.75 & 0.63 & 0.5 & 0.25 & 0.13 & 0.01 \end{bmatrix}$

- $P * 1e8$:

$$\begin{bmatrix} 124.94-3.93j & 9.2-3.83j & 3.53-3.53j & -0.-2.5j & -0.77-1.85j & -1.18-1.18j \\ 9.2-3.83j & 124.94-3.93j & 9.2-3.83j & 1.27-3.08j & -0.-2.5j & -0.77-1.85j \\ 3.53-3.53j & 9.2-3.83j & 124.94-3.93j & 3.53-3.53j & 1.27-3.08j & -0.-2.5j \\ -0.-2.5j & 1.27-3.08j & 3.53-3.53j & 124.94-3.93j & 9.2-3.83j & 3.53-3.53j \\ -0.77-1.85j & -0.-2.5j & 1.27-3.08j & 9.2-3.83j & 124.94-3.93j & 9.2-3.83j \\ -1.18-1.18j & -0.77-1.85j & -0.-2.5j & 3.53-3.53j & 9.2-3.83j & 124.94-3.93j \end{bmatrix}$$

- $P_B * 1e8$: $\begin{bmatrix} 1.27-3.08j & 3.53-3.53j & 9.2-3.83j & 9.2-3.83j & 3.53-3.53j & 1.27-3.08j \end{bmatrix}$

2.4 Section 4

2.4.1 Pseudocode

1. CALCULATE Contribution to Magnetic field density by unknown current sources (Q)
2. CALCULATE Contribution to Magnetic field density by current I_N (Q_B)

2.4.2 Code

```
# Q corresponds to contribution of unknown currents to H
Q = p.zeros((2*N-2,2*N-2),dtype = complex)
Q[:] = P[:]*a*(1/mu0)*((j*k/Ru[:]) + (1/(Ru[:]**2)))
print("Q : ",(Q).round(2))

# Qb corresponds to contribution of In to H
Qb = p.zeros((2*N-2,1), dtype = complex)
Qb[:,0] = Pb[:]*a*(1/mu0)*((j*k/Rz_del) + 1/(Rz_del**2))
print("Qb : ", (Qb).round(2))
```

2.4.3 Explanation

Q denotes the contribution of the unknown currents to Magnetic field density H. It is given by:

$$Q_{ij} = P_{ij}\frac{r}{\mu_o}(\frac{jk}{R_{ij}} + \frac{1}{R_{ij}^2})$$

Q_B denotes the contribution of I_N to the Magnetic field density. It is given by:

$$Q_B = P_B\frac{r}{\mu_o}(\frac{jk}{R_{iN}} + \frac{1}{R_{iN}^2})$$

2.4.4 Output Vectors (For $N = 4$)

- Q:

$$\begin{bmatrix} (99.521-0.001j) & (0.054-0.001j) & (0.008-0.001j) & (0.001-0.001j) & (0.001-0.001j) & -0.001j \\ (0.054-0.001j) & (99.521-0.001j) & (0.054-0.001j) & (0.003-0.001j) & (0.001-0.001j) & (0.001-0.001j) \\ (0.008-0.001j) & (0.054-0.001j) & (99.521-0.001j) & (0.008-0.001j) & (0.003-0.001j) & (0.001-0.001j) \\ (0.001-0.001j) & (0.003-0.001j) & (0.008-0.001j) & (99.521-0.001j) & (0.054-0.001j) & (0.008-0.001j) \\ (0.001-0.001j) & (0.001-0.001j) & (0.003-0.001j) & (0.054-0.001j) & (99.521-0.001j) & (0.054-0.001j) \\ -0.001j & (0.001-0.001j) & (0.001-0.001j) & (0.008-0.001j) & (0.054-0.001j) & (99.521-0.001j) \end{bmatrix}$$

- $Q_B : \begin{bmatrix} (0.003-0.001j) \\ (0.008-0.001j) \\ (0.054-0.001j) \\ (0.054-0.001j) \\ (0.008-0.001j) \\ (0.003-0.001j) \end{bmatrix}$

2.5 Section 5

2.5.1 Pseudocode

1. CALCULATE Current Density according to final equation by taking inverse ($\vec{J}$)
2. CALCULATE Current from $\vec{J}$ and boundary conditions (I)
3. CALCULATE Assumed value of current in dipole antenna ($I\_assumed$)
4. PLOT Current vs z for $N = 4$ and $N = 100$

2.5.2 Code

```
J = p.dot(p.inv(M(N)-Q),Qb)*Im
print("J : ", J)

# Finding current vector with boundary conditions
I[1:N] = J[:N-1,0]
I[N+1:-1] = J[N-1:,0]
print("I : ", I)

# Finding assumed current in the dipole antenna

I_assumed = p.zeros((2*N+1))
I_assumed[:] = Im*p.sin(k*(l-z[:]))

p.figure(1)
p.grid(True)
p.plot(z,abs(I), label = 'Actual Current')
p.plot(z,abs(I_assumed), label = 'Assumed Current')
```

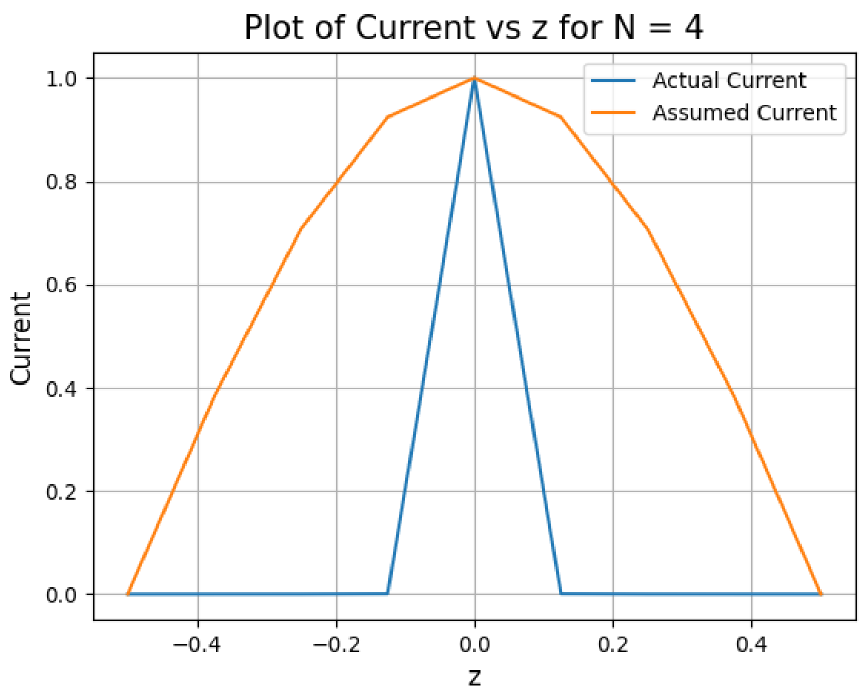
```
p.ylabel('Current',size = 12)
p.xlabel('z', size = 12)
p.title('Plot of Current vs z', size = 15)
p.legend(loc = 'upper right')
p.show()
```

2.5.3 Explanation and Plot

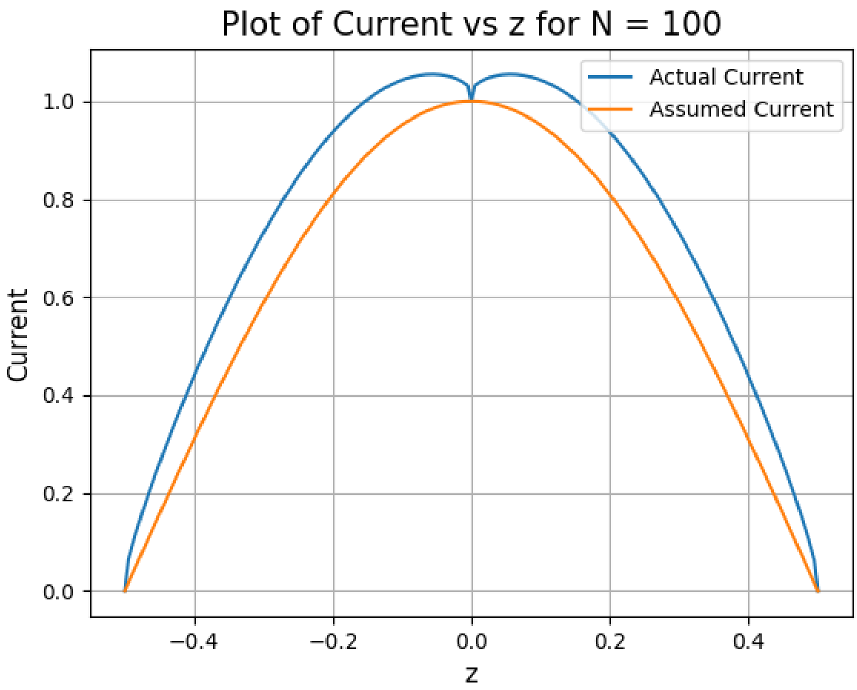
The final equation is:

$$(M - Q)J = Q_B I_m$$

This is solved to obtain $\vec{J}$ and then $\vec{I}$ is calculated using boundary conditions. The plot of current vs z for N = 4 is:



The plot for N = 100 is:



From the above plots, we can see that the calculated I is overshooting the assumed I curve at all points. This is due to **error in integration** which leads to error in the value of current

2.5.4 Output Vectors (For $N = 4$)

- J : $\begin{bmatrix} -3.30e-05+1.06e-05j \\ -9.55e-05+1.15e-05j \\ -6.48e-04+1.21e-05j \\ -6.48e-04+1.21e-05j \\ -9.55e-05+1.15e-05j \\ -3.30e-05+1.1e-05j \end{bmatrix}$

- I : $\begin{bmatrix} 0 \\ -3.30e-05+1.06e-05j \\ -9.55e-05+1.15e-05j \\ -6.48e-04+1.21e-05j \\ 1 \\ -6.48e-04+1.21e-05j \\ -9.55e-05+1.15e-05j \\ -3.30e-05+1.1e-05j \\ 0 \end{bmatrix}$

3 Conclusion

- Current profiles for wires in the antenna is calculated
- The assumed I is a very close approximation to the assumed result
- The overshooting of calculated I over the assumed I is due to error in integration. Integration here was not done perfectly. Hence it leads to errors in P, P_B, Q, Q_B and hence error in I