Assignment 9

Purvam Jain EE20B101
Electrical Engineering

May 13, 2022

Contents

1 Abstract

This is a problem to find the antenna currents in a half-wave dipole antenna A long wire carries a current $I(z)$ in a dipole antenna with half length of 50cm - so the antenna is a metre long, and has a wavelength of 2 metres. We want to determine the currents in the two wires of the antenna.The standard analysis assumes that the antenna current is given by

$$I = I_m sin(k(l-z)) \qquad 0 \leq z \leq l \tag{1}$$

$$I = I_m sin(k(l+z)) \qquad -l \leq z \leq 0 \tag{2}$$

We determine if this is a good assumption. To find the unknown current we get equations by calculating magnetic field by two different methods. First by Ampere's Law:

$$2\pi a H_\phi(z) = I_i$$

In Matrix form: $\begin{bmatrix} H_\phi[z1] \\ \dots \\ H_\phi[z_{N-1}] \\ H_\phi[z_{N+1}] \\ \dots \\ H_\phi[z_{2N-1}] \end{bmatrix} = \frac{1}{2\pi a} \begin{bmatrix} 1 & \dots & 0 & 0 & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & \dots & 1 & 0 & \dots & 0 \\ 0 & \dots & 0 & 1 & \dots & 0 \\ 0 & \dots & 0 & 0 & \dots & 1 \end{bmatrix} \begin{bmatrix} J_1 \\ \dots \\ J_{N-1} \\ J_{N+1} \\ \dots \\ J_{2N-1} \end{bmatrix}$

$$= M * J$$

The second computation involves the calculation of the vector potential:

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

This can be reduced to a sum as:

$$A_{z,i} = \sum_j I_j \left(\frac{\mu_o}{4\pi} \frac{exp(-jkR_{ij})}{R_{ij}} dz'_j \right)$$

$$= \sum_j P_{ij} I_j + P_b I_N$$

Now we know,

$$H_\phi(r,z) = \frac{-1}{\mu} \frac{\partial A_z}{\partial r}$$

$$H_\phi(r,z_i) = -\sum_j \frac{dz'_j}{4\pi} \left(\frac{-jk}{R_{ij}} - \frac{1}{R_{ij}^2} \right) exp(-jkR_{ij}) \frac{rI_j}{R_{ij}}$$

$$H_\phi(r,z_i) = \sum_j Q'_{ij} I_j = \sum_j Q_{ij} I_j + Q_{Bi} I_m$$

Using these equations we solve for J.

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

and hence can get I vector.

2 Program Structure

The program can be run from the command line as follows:

```
   python EE20B101.py -h
usage: EE20B101.py [-h] [-N N] [-r R] [-l L]

optional arguments:
  -h, --help  show this help message and exit
  -N N        Enter Number of Sections in each half section of the antenna.
  -r R        Enter radius of wire.
  -l L        Quarter Wavelength
```

3 Question 1

3.1 Pseudo Code#1

- Generate Array z = i*dz where -N¡=i¡=N ; points where we compute currents
- Use arange function to generate an array of 2N + 1 elements from -N to N
- Generate array u of 2N - 2 length of locations of unknown currents
- For this copy array z and remove known elements 0: at ends and Im at centre
- Construct the current vector I at points corresponding to vector z, and the current vector J at points corresponding to vector u.
- Reshape the array into column vectors.
- As instructed we use a if condition to print outputs only if N¡10.

We get the following output matrices for N=4:

```
   z:
 [[-0.5  ]
 [-0.375]
 [-0.25 ]
 [-0.125]
 [ 0.   ]
 [ 0.125]
 [ 0.25 ]
 [ 0.375]
 [ 0.5  ]]
```

```
u:
 [[-0.375]
 [-0.25 ]
 [-0.125]
 [ 0.125]
 [ 0.25 ]
 [ 0.375]]

I:
 [[0.]
 [0.]
 [0.]
 [0.]
 [1.]
 [0.]
 [0.]
 [0.]
 [0.]]

J:
 [[0.]
 [0.]
 [0.]
 [0.]
 [0.]
 [0.]]
```

4 Question 2

4.1 Pseudo Code#2

- Define a function to Generate M matrix
- We take N and r(radius) as inputs to this function
- Use identity function in numpy library to make identity matrix
- Print the matrix rounded to two decimal places if N,10
- Return M to user
- Implement the function on default values

We get the following output matrices for N=4:

```
M:
[[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]]
```

5 Question 3

5.1 Pseudo Code#3

- Define function to generate Ru and Rz matrices which takes inputs as radius, z and u vectors
- Define meshgrids on z and u vectors
- To compute distances Rij we take magnitude of resultant vector of R = r + z-z'
- Print rounded matrices if N¡10
- Return results upon calling function

We get the following output matrices for N=4:

```
Ru:
[[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]]

Rz:
[[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]]
```

5.2 Pseudo Code#4

- Define function to compute P and Pb matrices which take RiN vector and Ru matrix as input
- Implement formula to get Pb and P matrices
- Print the rounded matrices if N¡10 and return on implementation of function
- Define Rin vector as Nth column of Rz matrix without the known values of 0,2N and N indices
- Reshape the resultant array to get column vector
- Implement the earlier defined function to get P and Pb

We get th following values for P and Pb matrices:

```
   Pb:
[[1.27-3.08j]
[3.53-3.53j]
[9.2 -3.83j]
[9.2 -3.83j]
[3.53-3.53j]
[1.27-3.08j]]

P:
[[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]]
```

6 Question 4

6.1 Pseudo Code #5

- Define function to compute Qij and Qb matrices with inputs P,Pb,Rin,Ru and r as defined earlier
- Implement formula as given to get Qb and Q matrices

- Print of N¡10 and return the values to user

We get the Q and Qb matrices as follows:

```
   Qb:
[[0.00277-0.00089j]
[0.00802-0.00097j]
[0.05421-0.00101j]
[0.05421-0.00101j]
[0.00802-0.00097j]
[0.00277-0.00089j]]

Qij:
[[9.95209147e+01-0.00102798j 5.42084476e-02-0.00101222j
 8.02008034e-03-0.00096595j 1.24925031e-03-0.00079569j
 5.82883561e-04-0.00068249j 2.25969312e-04-0.0005595j ]
[5.42084476e-02-0.00101222j 9.95209147e+01-0.00102798j
 5.42084476e-02-0.00101222j 2.77231242e-03-0.0008922j
 1.24925031e-03-0.00079569j 5.82883561e-04-0.00068249j]
[8.02008034e-03-0.00096595j 5.42084476e-02-0.00101222j
 9.95209147e+01-0.00102798j 8.02008034e-03-0.00096595j
 2.77231242e-03-0.0008922j  1.24925031e-03-0.00079569j]
[1.24925031e-03-0.00079569j 2.77231242e-03-0.0008922j
 8.02008034e-03-0.00096595j 9.95209147e+01-0.00102798j
 5.42084476e-02-0.00101222j 8.02008034e-03-0.00096595j]
[5.82883561e-04-0.00068249j 1.24925031e-03-0.00079569j
 2.77231242e-03-0.0008922j  5.42084476e-02-0.00101222j
 9.95209147e+01-0.00102798j 5.42084476e-02-0.00101222j]
[2.25969312e-04-0.0005595j  5.82883561e-04-0.00068249j
 1.24925031e-03-0.00079569j 8.02008034e-03-0.00096595j
 5.42084476e-02-0.00101222j 9.95209147e+01-0.00102798j]]
```

7 Question 5

7.1 Pseudo Code#6

- Calculate J vector using inverse and matrix multiplication utilities of Numpy
- Insert the missing values of boundary conditions and current at centre
- Print rounded matrix if N¡10
- Concatenate the two-part original function into one original function
- Plot the original and estimated functions

The J matrix is as given below:

```
J:
[ 0.00000000e+00+0.00000000e+00j -3.30256482e-05+1.06463792e-05j
-9.54636142e-05+1.15207845e-05j -6.48254232e-04+1.20785421e-05j
 1.00000000e+00+0.00000000e+00j -6.48254232e-04+1.20785421e-05j
-9.54636142e-05+1.15207845e-05j -3.30256482e-05+1.06463792e-05j
 0.00000000e+00+0.00000000e+00j]
```

8 Outputs and Conclusions

The following graphs are obtained:

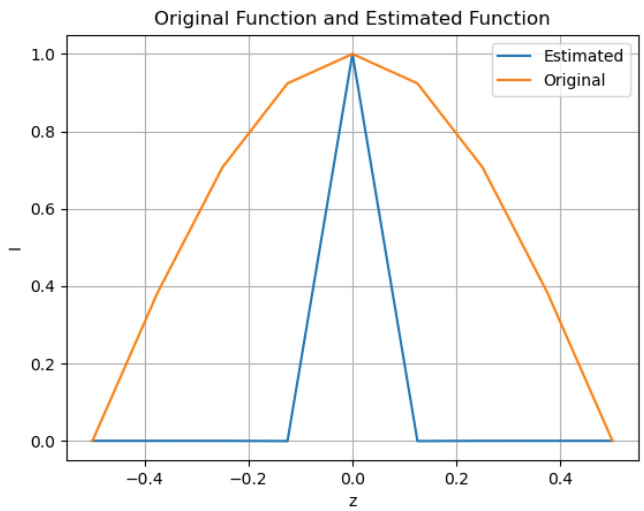

Figure 1: Current vs z for N=4

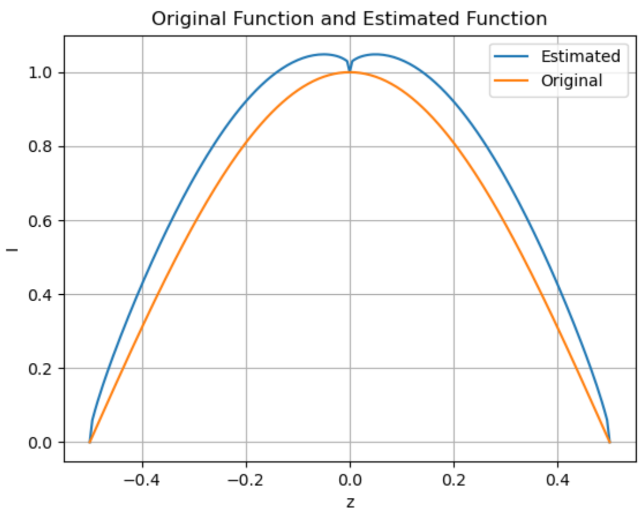

Figure 2: Current vs z for N=100

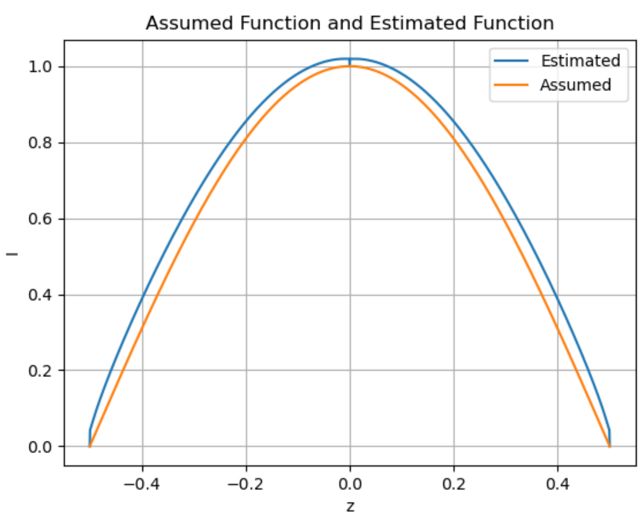

Figure 3: Current vs z for N=800

As expected the results are quite close to a sinusoid. The current on a half-wave dipole antenna is approximately sinusoidal with a node at each end and anti-node at the centre(feed point). This current distribution looks like a standing wave. The discrepancies observed are due to less number of sections considered along the length of dipole. As the number of sections(N) is increased the error in the estimated and assumed functions decreases.The kink at the top is due to the enforced boundary condition of feeder current being Im at that point.