EE2703: ENDSEMESTER EXAMINATION

ALLU YASWANTH, EE20B007

May 12, 2022

1 Abstract

- To find the antenna currents in a half-wave dipole antenna using:

  (i) Standard Expression. (ii) Magnetic Vector Potential and approximating

- To study the difference between the graphs obtained via estimation and actual values

2 Introduction

We have a long wire carrying a current I(z) in dipole antenna with half length 0f 50cm(=l) so, wavelength = 2m. Next, we need to determine the currents in the two wires of the antenna. Next, we have the expressions to calculate the value of currents.

$$I = I_m sin(k(l - z))$$
$$0 \le z \le l$$
$$I = I_m sin(k(l + z))$$
$$-l \le z < 0$$

In the next process, we calculate the magnetic vector potential by approximating the integrals (in terms of summation); we next find out P_{ij} and P_B.

$$A_{z,i} = \sum_j P_{ij} I_j + P_B I_N = \sum_j I_j (\frac{\mu_0}{4\pi} \frac{exp(-jkR_{ij})}{R_{ij}} dz'_j)$$

$$P_B = \frac{\mu_0}{4\pi} \frac{exp(-jkR_{iN})}{R_{iN} dz'_j}$$

Then, we use the Ampere's circuital law to calculate H_ϕ. Again, we get it in terms of some summation involving the matrices Q_{ij} and Q_B.

$$H_\phi(r, z_i) = \sum_j Q_{ij} J_j + Q_{Bi} I_m = -\sum_j P_{ij} \frac{r}{\mu_0} (\frac{-jk}{R_{ij}} - \frac{1}{R_{ij}^2}) + P_B \frac{r}{\mu_0} (\frac{-jk}{R_{iN}} - \frac{1}{R_{iN}^2})$$

At last we solve the matrix equation to find out the current vector J and then find out I.

$$MJ = QJ + Q_B I_m$$

3 Assignment questions

3.1 Question 1

According to the question, we now need to find vector z and u. And then find the current vectors I (at loactions of z) and J (at locations of u) respectively.

The following code snippet does the job!

```
#Question 1
z = arange(-len, len+len/N , len/N) #Points in A.P where we compurte the currents
I = np.zeros(2 * N + 1) # Initiating the I matrix with zeros
for i in range(0,N):
    I[i] = Im * sin(k * (len + z[i])) #Giving the I values as given in the problem
for i in range(N,2*N):
    I[i] = Im * sin(k * (len - z[i]))  #Giving the I values as given in the problem

u = [(i*dz)-0.5 for i in range(1, 2 * N)]
u.__delitem__(N-1 ) #u gives the matrix z excluding the edge points and the middle one.
```

```
The values obtained after running the code are:

z = [-0.5   -0.38  -0.25  -0.12   0.     0.12   0.25   0.38   0.5 ]
I = [0.    0.38  0.71  0.92  1.    0.92  0.71  0.38  0.   ]
u = [-0.38  -0.25  -0.12   0.12   0.25   0.38]
```

3.2 Question 2

According to the question, we now need determine the M vector.I defined a function to determine M vector

The following code snippet does the job!

```
def Matrix_M():
    M = np.identity(2 * N - 2)
    M = (1 / (2 * PI * rad)) * M
    return M
```

```
The values obtained after running the code are:
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]]
```

3.3 Question 3

We will determine Rz,Ru,P,PB in this question using the formulas given the assignment.
The following code snippet does the job!

```
#Question 3
#Determining Rz
Z = np.meshgrid(z,z)
Z_i = Z[0]
Z_j = Z[1]
Rz = np.sqrt((Z_i-Z_j)**2 + np.ones([2*N+1,2*N+1],dtype=complex)*(rad**2))

#Determining Ru
U = np.meshgrid(u,u)
U_i = U[0]
U_j = U[1]
Ru = np.sqrt((U_i-U_j)**2 + np.ones([2*N-2,2*N-2],dtype=complex)*(rad**2))

Rin = Rz[N]
Rin = np.delete(Rin, [0, N, 2 * N], 0)

#Computing the vectors P and PB
# P gives the Contribution due to all currents
P = np.zeros((2 * N - 2, 2 * N - 2), dtype=complex)
for i in range(2 * N - 2):
    for j in range(2 * N - 2):
        P[i][j] = (mu0 / (4.0 * PI)) * (np.exp(-1j * k * Ru[i][j])) * dz / Ru[i][j]

# Contribution of vector potential due to current IN
PB = (mu0 / (4 * PI)) * (np.exp(-1j * k * Rin)) * dz / Rin
```

```
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]]
```

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

3.4 Question 4

According to the question, we now need determine Q and QB using the formulas given in the assignment.
The following code snippet does the job!

```
#Question 4
# Computing Q and QB from given formula
Q = np.zeros((2 * N - 2, 2 * N - 2), dtype=complex)
for i in range(2 * N - 2):
    for j in range(2 * N - 2):
        Q[i][j] = -P[i][j] * (rad / mu0) * ((-1j * k / Ru[i][j]) - (1 / pow(Ru[i][j], 2)))


QB = -PB * (rad / mu0) * ((-1j * k / Rin) - (1 / Rin ** 2))
```

```
Q:    [[9.952e+01-0.j  5.000e-02-0.j  1.000e-02-0.j  0.000e+00-0.j  0.000e+00-0.j
  0.000e+00-0.j]
 [5.000e-02-0.j  9.952e+01-0.j  5.000e-02-0.j  0.000e+00-0.j  0.000e+00-0.j
  0.000e+00-0.j]
 [1.000e-02-0.j  5.000e-02-0.j  9.952e+01-0.j  1.000e-02-0.j  0.000e+00-0.j
  0.000e+00-0.j]
 [0.000e+00-0.j  0.000e+00-0.j  1.000e-02-0.j  9.952e+01-0.j  5.000e-02-0.j
  1.000e-02-0.j]
 [0.000e+00-0.j  0.000e+00-0.j  0.000e+00-0.j  5.000e-02-0.j  9.952e+01-0.j
  5.000e-02-0.j]
 [0.000e+00-0.j  0.000e+00-0.j  0.000e+00-0.j  1.000e-02-0.j  5.000e-02-0.j
  9.952e+01-0.j]]
Q_B:    [0.    -0.j  0.01-0.j  0.05-0.j  0.05-0.j  0.01-0.j  0.    -0.j]
```

3.5 Question 5

We will determine estimated current in this question and plot a graph comparing estimated and assumed currents. Estimated currents are further determined by imposing boundary conditions.
The following code snippet does the job!

```
#Question 5
result_J = np.dot(linalg.inv(M - Q), QB) #The resultant I vector which doesn't include the e
result_I = np.zeros(2*N+1, dtype = complex)#Defining a zero array to store the I values
#Forming the I vector for all elements by adding boundary conditions
result_I[1:N] = result_J[0:N-1]
result_I[N+1: 2*N] = result_J[N-1:2*N-1]
result_I[N] = Im  # I at z=0 is Im
#I at edges is 0
```

```
I estimated:  [ 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]
```

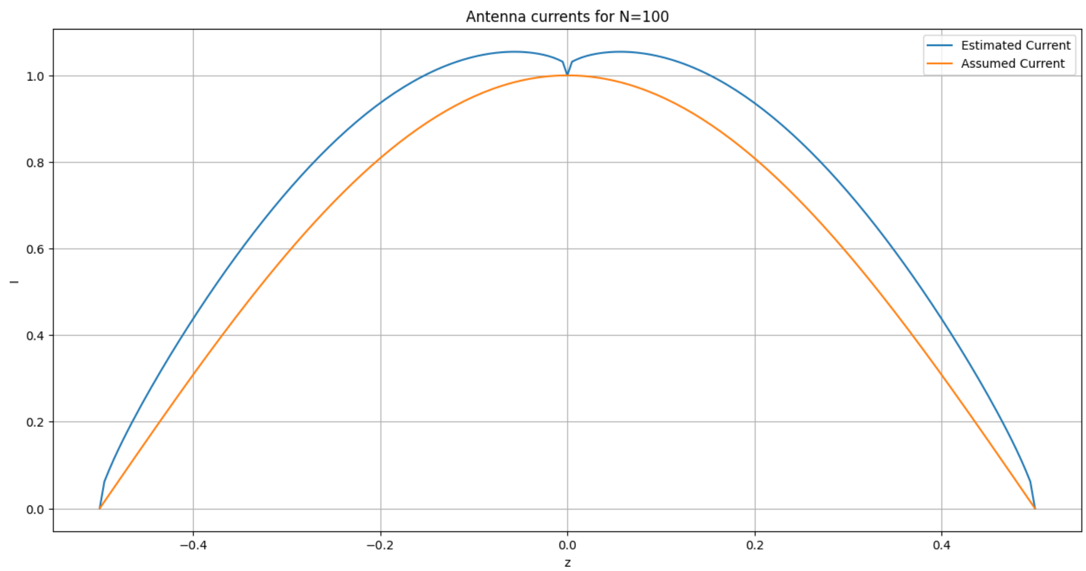

Figure 1: Plot of assumed current Vs estimated current

4 CONCLUSION

- On increasing the value of N,the both graph will merge each other.
- On increasing N,the magnitude of point which are away from the centre are increasing.