EE2703 APL : Final Exam

Sagar (EE20B115)

May 12, 2022

1   Aim

In this assignment we aim to analyse if the current in antenna calculated using the standard equation agrees with current calculated using different method as shown in assignment.

2   Procedures and Observations

2.1   Problem Statement 1

In this section we create vector I and z which hold the currents (known and unknown) in antenna elements and their locations respectively. Also we have to create vectors J and u which will hold only the unknown values and their locations.

Expression used:

$$z = i \times dz, -N \leq i \leq N \tag{1}$$

2.1.1   Code snippet:

```
z  = np.linspace(-N,N,TotalCurrentElements)
z = z*l/N

I = np.zeros(TotalCurrentElements)
I[0] = 0
I[N] = Im
I[2*N] = 0

u = np.delete(z,[0,N,2*N])
J = np.zeros(2*N-2)
```

2.1.2 Observations

Following arrays/vectors are obtained on running the code:

$$z = \begin{bmatrix} 0.5 \\ 0.38 \\ 0.25 \\ 0.12 \\ 0.0 \\ -0.12 \\ -0.25 \\ -0.38 \\ -0.5 \end{bmatrix} \quad I = \begin{bmatrix} 0.0 \\ 0.0 \\ 0.0 \\ 0.0 \\ 1.0 \\ 0.0 \\ 0.0 \\ 0.0 \\ 0.0 \end{bmatrix} \quad u = \begin{bmatrix} 0.38 \\ 0.25 \\ 0.12 \\ -0.12 \\ -0.25 \\ -0.38 \end{bmatrix} \quad J = \begin{bmatrix} 0.0 \\ 0.0 \\ 0.0 \\ 0.0 \\ 0.0 \\ 0.0 \end{bmatrix}$$

2.2 Problem Statement 2

In this section we aim to create a function that accepts N as an input and return a diagonal matrix with dimensions N-2 by N-2.
Expressions used:

$$2\pi H_\phi(z_i) = I_i \tag{2}$$

and

$$H_\phi = M \times J \tag{3}$$

where **M** is a diagonal matrix.

2.2.1 Code snippet:

```
def matrixM(N):
M = np.zeros((2*N−2,2*N−2), dtype=float)
np.fill_diagonal(M,1)
return (1/(2*PI*a))*M
```

2.2.2 Observations

For N=4, we obtain the following matrix M:

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

2.3 Problem Statement 3

In this section we aim to create a function that accepts N as an input and return a diagonal matrix with dimensions N-2 by N-2.
Expressions used:

$$P_{ij} = \frac{\mu_0}{4\pi} \frac{exp(-jkR_{ij})}{R_{ij}} dz \tag{4}$$

2

Also

$$P_B = \frac{\mu_0}{4\pi} \frac{exp(-jkR_{iN})}{R_{iN}} dz \qquad (5)$$

2.3.1 Code snippet:

```
R = np.zeros((2*N+1,2*N+1),dtype=float)

zy,zx = np.meshgrid(z,z)
R = np.sqrt(a**2 + (zx-zy)**2)

X = np.copy(R)
X = np.delete(X,[0,N,2*N],axis=1)

R1n = X[N,:]
Pb = (((mu*dz)/(4*PI))*(np.exp(-1j*k*R1n)))/R1n

X = np.delete(X,[0,N,2*N],axis=0)

P = np.zeros((2*N-2,2*N-2), dtype= complex)
P = ((mu*dz/(4*PI))*np.exp(-1j*k*X))/X
```

2.3.2 Observations

For N=4, we obtain the following P and P_B matrices:

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

$$P_B = \begin{bmatrix} 1.27 - 3.08j \\ 3.53 - 3.53j \\ 9.2 - 3.83j \\ 9.2 - 3.83j \\ 3.53 - 3.53j \\ 1.27 - 3.08j \end{bmatrix}$$

2.4 Problem Statement 4

In this section we take the curl of vector potential and find H field which is later made equal to H field obtained earlier. Using that we obtain Q and Q_B matrices where Q_B is a column vector.
Expressions used:

$$Q_{ij} = P_{ij} \frac{r}{\mu_0} (\frac{jk}{R_{ij}} + \frac{1}{R_{ij}^2}) \qquad (6)$$

3

Also

$$Q_B = P_B \frac{r}{\mu_0}(\frac{jk}{R_{iN}} + \frac{1}{R_{iN}^2}) \tag{7}$$

2.4.1   Code snippet:

```
Q = np.zeros((2*N-2,2*N-2), dtype=complex)

Q = (a/mu)*P*(((1j*k)/X)+(1/(X**2)))

Qb = (a/mu)*Pb*((1j*k/R1n)+(1/(R1n**2)))
```

2.4.2   Observations

For N=4, we obtain the following Q and Q_B matrices:

$$Q = \begin{bmatrix} 99.52 & 0.05 & 0.01 & 0.0 & -0.0 & 0.0 \\ 0.05 & 99.52 & 0.05 & 0.0 & 0.0 & 0.0 \\ 0.01 & 0.05 & 99.52 & 0.01 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.01 & 99.52 & 0.05 & 0.01 \\ 0.0 & 0.0 & 0.0 & 0.05 & 99.52 & 0.05 \\ 0.0 & 0.0 & 0.0 & 0.01 & 0.05 & 99.52 \end{bmatrix}$$

$$Q_B = \begin{bmatrix} 0.0 \\ 0.01 \\ 0.05 \\ 0.05 \\ 0.01 \\ 0.0 \end{bmatrix}$$

2.5   Problem Statement 5

In this section we have to solve for **J** using the equation given below. **J** can be obtained by multiplying both sides with inverse of **M-Q** as done below.
Expressions used:

$$MJ = QJ + Q_B I_m \tag{8}$$

or equivalently

$$(M - Q)J = Q_B I_m \tag{9}$$

Then **I** is obtained using **J** by assigning corresponding elements.

2.5.1   Code snippet:

```
J = np.dot(np.linalg.inv(M-Q),Qb)

I[1:N]    = abs(J[0:N-1])
I[N+1:-1] = abs(J[N-1:])
```

2.5.2 Observations

Values of **I** and **J** are not meaningful for such small value of N.

3 Plot of I vs l for N = 100

Now we change the value of N from 4 to 100 and Plot a graph of Current in Antenna **I** vs Length of Antenna **l** as shown:
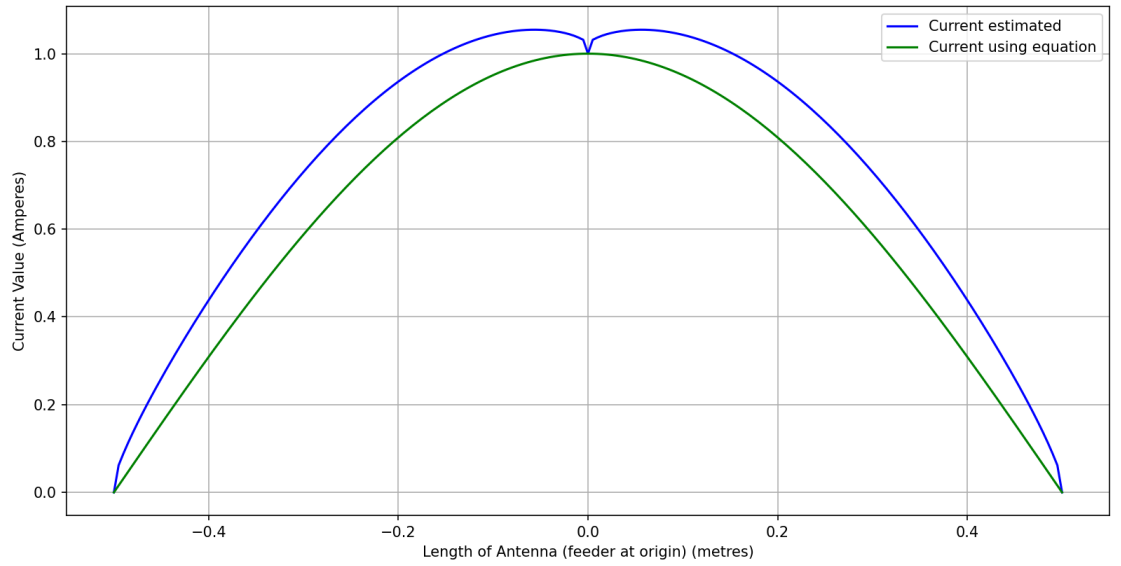
3.0.1 Plot

Figure 1: Plot of I vs l

3.0.2 Observations and Conclusions

We can make the following observations from the plot:

1. The shape of both graphs is approximately the same.

2. The current estimated is slightly higher than the one obtained using the equation.

This minute offset that is observed in the plot arises because we have take a finite number of elements while doing our estimation. Which is not the case for the formula as it is obtained using integration. Dues to this limitation, the offset arises in plots.