Half wave dipole antenna current distribution

Kathir Pagalavan EE20B056

May 12, 2022

1 Aim

To validate the assumption of sinusoidal current in an half-wave dipole antenna by an FEM analysis using Ampere's law and magnetic vector potential equations.

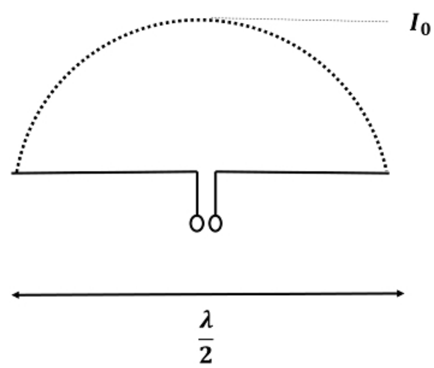

Figure 1: Sinusoidal current distribution assumption

2 Splitting into elements

We'll split each half of the Antenna into N elements going from 0 to 2N(since python lists cannot have negative indices).We compute the currents at each of the edge points of these elements. The only current we know are the zeroth(Nth in list).Entering current through one half of the antenna and leaving current through the other half of the antenna can be modeled as a single element with current Im.

$$I(z) = \left\{ \begin{array}{ll} I_m sin(k(l-z)), & 0 \leq z \leq l \\ I_m sin(k(l+z)), & -l \leq z \leq 0 \end{array} \right\}$$

```
l=0.5           #quarter wavelength
c=2.9979e8      #speed of light
```

```
mu0=4e-7*pi       #permeability of free space
Im=1              #current injected into the antenna
a=0.01            #radius of wire
lamda=l*4         #wavelength
f=c/lamda         #frequency
k=2*pi/lamda      #wavenumber

I=zeros(2*N+1)  #'I' vector representing all currents
I[0]=I[2*N]=0;I[N]=Im   #inserting boundary values in 'I' vector
I[1:N],I[N+1:2*N]=real(J[0:N-1]),real(J[N-1:])  #inserting the unknown currents in the 'I'
```

```
PSEUDOCODE
split I
insert boundary values
split J and insert into I
```

3 Unknown currents in terms of Ampere's law

$$2\pi a H_\Phi = I_i$$

writing this as a matrix equation

$$\begin{pmatrix} H_\phi[z_1] \\ \dots \\ H_\phi[z_{N-1}] \\ H_\phi[z_{N+1}] \\ \dots \\ H_\phi[z_{2N-1}] \end{pmatrix} = \frac{1}{2\pi a}\begin{pmatrix} 1 & \dots & 0 & 0 & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & \dots & 1 & 0 & \dots & 0 \\ 0 & \dots & 0 & 1 & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & \dots & 0 & 0 & \dots & 1 \end{pmatrix}\begin{pmatrix} J_1 \\ \dots \\ J_{N-1} \\ J_{N+1} \\ \dots \\ J_{2N-1} \end{pmatrix}$$

$$H = MJ$$

Note that 0, N and 2N elements were removed, making the matrix of size 2N-2.

```
def M(N):    #function to create M matrix
    return identity(2*N-2)/(2*pi*a)
```

From above code,for N=4, we get

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

4 Unknown currents in terms of magnetic vector potential

$$A_{rz} = \frac{\mu_0}{4\pi}\int \frac{I_z e^{-jkR}}{R} dz$$

Writing this in matrix form, the contribution of jth current element to ith observer is given by

$$P_{ij} = \frac{\mu_0}{4\pi}\frac{e^{-jkR_{ij}}}{R_{ij}} dz$$

Forming an array of P_{ij} without 0,N,2N elements would give a matrix of size 2N-2 , P. Contribution of the Nth element alone to the i^{th} observer can be put as a column vector of size 2N-2, P_B.From this, contribution of each of these elements to the H vector can be given as

$$Q_{ij} = P_{ij}\frac{\mu_0}{4\pi}(\frac{jk}{R_{ij}} + \frac{1}{R_{ij}^2})$$

```
def consmat(N): #function to CONStruct the MATrices Q and Qb

    Rz,Ru,Rn=R(N)
    dz=l/N

    Pb=mu0*dz/(4*pi)*exp(-1j*k*Rn)/Rn
    Qb=a*(1j*k*divide(Pb,Rn)+divide(Pb,square(Rn)))/mu0

    P=mu0*dz*divide(exp(complex(0,-k)*Ru),Ru)/(4*pi)
    Q=a*(complex(0,k)*divide(P,Ru)+divide(P,square(Ru)))/mu0

    return(Q,Qb)
```

From the above code, for N=4, we get

$$Pb = \begin{pmatrix} 1.27-3.08j \\ 3.53-3.53j \\ 9.2-3.83j \\ 9.2-3.83j \\ 3.53-3.53j \\ 1.27-3.08j \end{pmatrix}$$

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

$$Qb = \begin{pmatrix} 2.77 - 0.89j \\ 8.02 - 0.97j \\ 54.21 - 1.01j \\ 54.21 - 1.01j \\ 8.02 - 0.97j \\ 2.77 - 0.89j \end{pmatrix}$$

Pb and P have been adjusted for comfortable visibility by multiplying with 1e8 and Qb has been multiplied with 1e3, but the Q matrix as a whole has wide range so couldn't be adjusted to put into a decimal matrix here.

In the above implementation, the contribution to A and H by Nth element are given as the vectors P_B and Q_B respectively and the contribution to the same by each of the rest of elements is given as matrices P and Q respectively.

We arrive at R_{ij} by calculating all the distances between jth current element and ith observer position.We achieve it by using *meshgrid* to form z coordinate differences of both the set of points and then use it to calculate the distances at once.

```
def R(N):   #function returning distance between points
    t=linspace(0,2*N,2*N+1)
    x,y=meshgrid(t,t)
    Rz=sqrt(((l/N)*(x-y))**2+a**2)    #distance between all sets of points
    Rn=concatenate( ( (Rz[:,N])[1:N], (Rz[:,N])[N+1:2*N] ) )    #distance between Im current
    Ru=delete(Rz,[0,N,-1],0);Ru=delete(Ru,[0,N,-1],1)   #distance between all sets of points
    return(Rz,Ru,Rn)
```

```
PSEUDOCODE
array [0,2N]
x,y = meshgrid with itself
x-y #number of splits gap between all source and observers
z gap = (x-y)*dz
sqrt(zgap**2+radius**2)
```

From the above code, for N=4, we get

$$R_z = \begin{pmatrix} 0.01 & 0.13 & 0.25 & 0.38 & 0.5 & 0.63 & 0.75 & 0.88 & 1. \\ 0.13 & 0.01 & 0.13 & 0.25 & 0.38 & 0.5 & 0.63 & 0.75 & 0.88 \\ 0.25 & 0.13 & 0.01 & 0.13 & 0.25 & 0.38 & 0.5 & 0.63 & 0.75 \\ 0.38 & 0.25 & 0.13 & 0.01 & 70.3 & 0.25 & 0.38 & 0.5 & 0.63 \\ 0.5 & 0.38 & 0.25 & 0.13 & 0.01 & 0.13 & 0.25 & 0.38 & 0.5 \\ 0.63 & 0.5 & 0.38 & 0.25 & 0.13 & 0.01 & 0.13 & 0.25 & 0.38 \\ 0.75 & 0.63 & 0.5 & 0.38 & 0.25 & 0.13 & 0.01 & 0.13 & 0.25 \\ 0.88 & 0.75 & 0.63 & 0.5 & 0.38 & 0.25 & 0.13 & 0.01 & 0.13 \\ 1. & 0.88 & 0.75 & 0.63 & 0.5 & 0.38 & 0.25 & 0.13 & 0.01 \end{pmatrix}$$

5 Comparing matrices

By comparing both the H matrices we can arrive at the unknown currents as follows.

$$MJ = QJ + Q_B I_m$$

$$J = (M - Q)^{-1} Q_B Im$$

Inserting the boundary conditions $I[0] = I[2N] = 0$ and $I[N] = I_m$ to the J vector we arrive at the current distribution on the half-wave dipole antenna.

```
N=100
Q,Qb=consmat(N)
t=linspace(0,2*N,2*N+1)-(N)*ones(2*N+1)
Iexp=Im*sin(k*(l-(l/N)*abs(t)))
J=(matmul(inv(M(N)-Q),Qb)*Im)   #J vector representing unknown currents
I=zeros(2*N+1)  #'I' vector representing all currents
I[0]=I[2*N]=0;I[N]=Im   #inserting boundary values in 'I' vector
I[1:N],I[N+1:2*N]=real(J[0:N-1]),real(J[N-1:])  #inserting the unknown currents in the 'I'
```

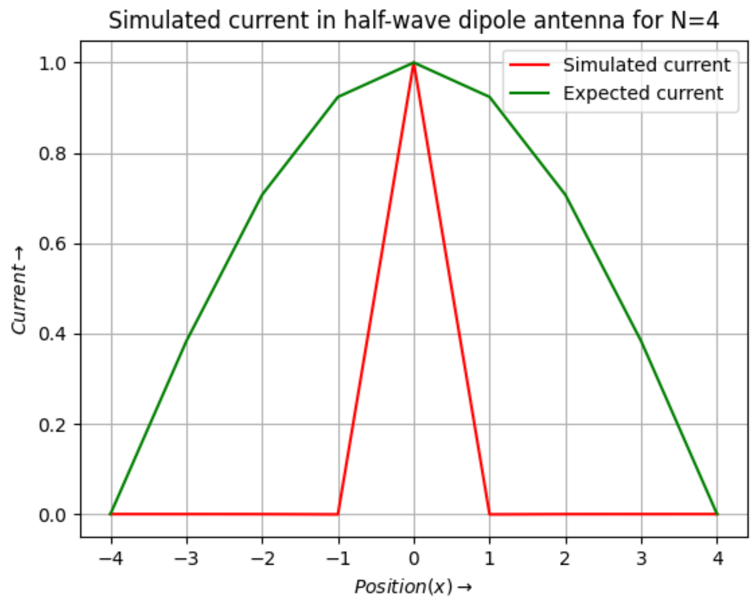

Figure 2: Simulated current distribution

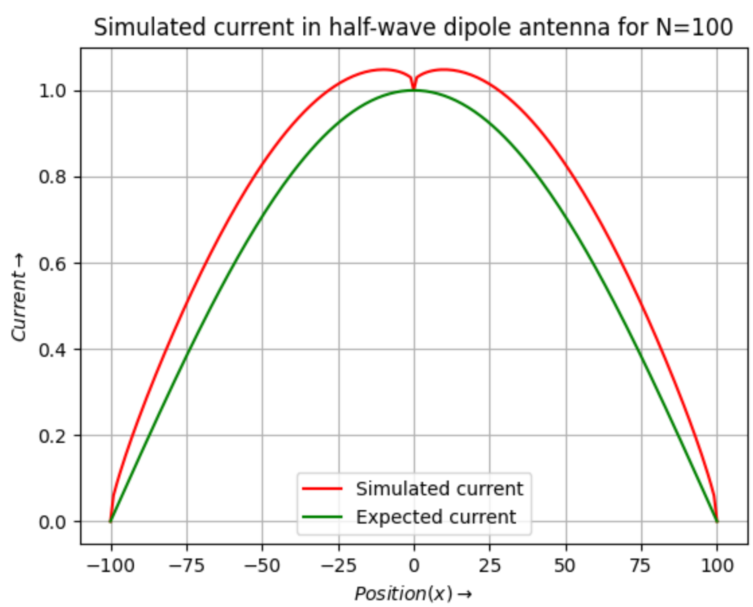

Figure 3: Simulated current distribution