Final Exam for EE2703 (20 Marks) **May 02, 2022**

Dept. of Electrical Engineering, IIT Madras
Applied Programming Lab Jan 2021 session

- ▷ **This code involves a lot of vectors and arrays. All such operations should be vectorized.**
- ▷ **Label all plots. Add legends. Make the plots professional looking.**
- ▷ **Comments are not optional. They are required.**
- ▷ **There are five sections to the code. Each should have its own pseudocode (you can cut and paste the question itself here to some extent)**
- ▷ **I expect to see each pseudocode should be readable and neatly formatted.**
- ▷ **PDF file should be named *your-roll-number.pdf***
- ▷ **Each array or vector asked for in this assignment should be printed out in the pdf. For this, use $N = 4$. The printing of vectors and arrays is messy so use print((MATRIX).round(2)), where MATRIX is the matrix (or vector) you are printing.**
- ▷ **Only in the case of the matrix P, use "print((P*1e8).round(2))" so that the numbers are meaningful after rounding to two digits.**
- ▷ **Once you have the code debugged, set N=100 and do the actual calculation (do not print out all the matrices here!)**
- ▷ **Python code should be named *your-roll-number.py* Please note that I will accept only raw python code and it should run in Python 3.x (prefer Python 3.8) So don't send me Jupyter notebooks.**
- ▷ **Python code should run!!**
- ▷ **Pdf file should include all plots and tables.**
- ▷ **The Pdf should be submitted to the 'final' assignment, and the .py code should be submitted to the 'final-code' assignment.**

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

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

This is then used to compute the radiated field. The problem is to determine if this is a good assumption. The parameters of this problem are as follows:

l=0.5 metres (quarter wavelength) c=2.9979e8 m/sec, speed of light mu0=4e-7*pi permeability of free space N=4 Number of sections in each half section of the antenna. Set to N=4 initially and to N=100 later. Im=1.0 current injected into the antenna. a=0.01 metres, radius of wire. Dependent Parameters lamda=l*4.0 metres, wavelength f=c/lamda Hz, frequency k=2*pi/lamda wave number dz=l/N spacing of current samples

1. **Divide the wire into pieces of length dz. Ideally we should number the pieces with indices going from $-N$ to $+N$. Unfortunately, Python does not allow negative array indices, so we will have an array with indices going from 0 to $2N$ ($2N+1$ elements, with element N being the feed of the antenna).**

   Define points along the antenna as an array, z,

   $$z = i \times dz, \quad -N \leq i \leq N$$

   These are the points at which we compute the currents. The currents at end of the wire are zero, while the currents at $z=0$ are prescribed by the circuit driving the antenna. There is the entering current I_m and the return current $-I_m$. Both currents are at $z=0$, and both point in the same direction along the antenna. So a single value is sufficient. The currents are therefore $I[i]$ for $0 \leq i \leq N$. So there are $2N+1$ currents, with $2N-2$ currents unknown (The end currents are known to be zero and current at the centre is given as I_m.) The $2N-2$ locations of unknown currents are computed and kept in array u. Also construct the current vector I at points corresponding to vector z, and the current vector J at points corresponding to vector u.

2. **We need an equation for each unknown current. These equations are obtained by calculating the Magnetic field in two different ways. From Ampere's Law, we have for $H_\phi(z, r=a)$**

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

write this as a matrix equation

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

Note that the matrix M assumes H_ϕ is computed at $r = a$. Also note that the vector J is the vector of unknown currents. This is why the matrix is $2N-2$ by $2N-2$.

Write a function to compute and return the matrix M.

3. **The second computation involves the calculation of the vector potential**

$$\vec{A}(r,z) = \frac{\mu_0}{4\pi}\int \frac{I(z')\hat{z}e^{-jkR}dz'}{R}$$

where $\vec{R} = \vec{r} - \vec{r}' = r\hat{r} + z\hat{z} - z'\hat{z}$ and $k = \omega/c = 0.1$. $\vec{r}$ is the point where we want the field, and $\vec{r}' = z'\hat{z}'$ is the point on the wire. This can be reduced to a sum:

$$\begin{aligned} A_{z,i} &= \frac{\mu_0}{4\pi}\sum_j \frac{I_j \exp\left(-jkR_{ij}\right) dz'_j}{R_{ij}} && (1) \\ &= \sum_j I_j \left(\frac{\mu_0}{4\pi}\frac{\exp(-jkR_{ij})}{R_{ij}} dz'_j\right) && (2) \\ &= \sum_j P_{ij} I_j + P_B I_N && (3) \end{aligned}$$

again a matrix equation like the previous one. P is a matrix with $2N-2$ columns and $2N-2$ rows. Note that the vector potential is driven by all the currents, which is why we use the I vector. P_B is the contribution to the vector potential due to current I_N, and is given by

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

Compute and create vectors Rz and Ru which are the distances from observer at $\vec{r} + z_i\hat{z}$, and source at $z'_j\hat{z}$. The difference between Rz and Ru is that the former computes distances including distances to known currents, while Ru is a vector of distances to unknown currents.

Also compute the matrix P and P_B . Note that P_B is a column vector.

We only want the ϕ component of $\vec{H}$ and $\vec{A}$ only has the $\hat{z}$ component. So the equation become

$$H_\phi(r,z) = -\frac{1}{\mu}\frac{\partial A_z}{\partial r} = -\sum_j \frac{\mu_0}{4\pi}\frac{dz'_j}{\mu}\frac{\partial}{\partial r}\left(\frac{\exp\left(-jkR_{ij}\right)}{R_{ij}}\right) I_j$$

Note that all the currents contribute to H_ϕ, and so the current vector is I. Now, $\vec{R} = r\hat{r} + (z-z')\hat{z}$. So, $R = \sqrt{r^2 + (z-z')^2}$. The derivative becomes

$$\frac{\partial}{\partial r} R_{ij} = \frac{1}{2R_{ij}} 2r = \frac{r}{R_{ij}}$$

So,

$$\begin{aligned} H_\phi(r,z_i) &= -\sum_j \frac{dz'_j}{4\pi}\left(\frac{-jk}{R_{ij}} - \frac{1}{R_{ij}^2}\right)\exp\left(-jkR_{ij}\right)\frac{rI_j}{R_{ij}} \\ &= -\sum_j P_{ij}\frac{r}{\mu_0}\left(\frac{-jk}{R_{ij}} - \frac{1}{R_{ij}^2}\right) I_j + P_B\frac{r}{\mu_0}\left(\frac{-jk}{R_{iN}} - \frac{1}{R_{iN}^2}\right) I_m \\ &= \sum_j Q'_{ij} I_j \end{aligned}$$

We now have a second expression for $H_\phi(r,z_i)$:

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

The Q'_{ij} in the equation is over all the currents. However this needs to be split into the unknown currents, J_j and the boundary currents. Only one of the boundary currents is non-zero, namely the feed current at $i = N$. The matrix corresponding to J_j we call Q_{ij}, and the boundary current we call $Q_B = Q'_{iN}$

Create the matrices Q_{ij} and Q_B.

5. **Our final equation is**

$$MJ = QJ + Q_B I_m$$

i.e.,

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

This is easily solved for to obtain $\vec{J}$, and hence $\vec{I}$. The current vector can be compared to the standard expression given at the top of the assignment.

Invert $(M-Q)$ and obtain J. Use *inv(M-Q)* in python. Add the Boundary currents (zero at i=0, i=2N, and I_m at i=N). Then plot this current vs. z and also plot the equation assumed for current at the top of this question paper.

Explain any discrepancies.

Useful Python Commands (use "?" to get help on these from ipython)

```
from pylab import *
import system-function as name
Note: lstsq is found as scipy.linalg.lstsq
ones(List)
zeros(List)
range(N0,N1,Nstep)
arange(N0,N1,Nstep)
linspace(a,b,N)
logspace(log10(a),log10(b),N)
X,Y=meshgrid(x,y)
where(condition)
where(condition & condition)
where(condition | condition)
a=b.copy()
lstsq(A,b) to fit A*x=b
A.max() to find max value of numpy array (similalry min)
A.astype(type) to convert a numpy array to another type (eg int)
def func(args):
  ...
  return List
matrix=c_[vector,vector,...] to create a matrix from vectors
figure(n) to switch to, or start a new figure labelled n
plot(x,y,style,...,lw=...)
semilogx(x,y,style,...,lw=...)
semilogy(x,y,style,...,lw=...)
loglog(x,y,style,...,lw=...)
```

```
contour(x,y,matrix,levels...)
quiver(X,Y,U,V) # X,Y,U,V all matrices
xlabel(label,size=)
ylabel(label,size=)
title(label,size=)
xticks(size=) # to change size of xaxis numbers
yticks(size=)
legend(List) to create a list of strings in plot
annotate(str,pos,lblpos,...) to create annotation in plot
grid(Boolean)
show()
```