Final Exam for Group A (20 Marks) **Apr 19, 2017**

Dept. of Electrical Engineering, IIT Madras
Applied Programming Lab Jan 2017 session

- **Time duration of exam is 2.5 hours**
- **Create a work directory called *your-roll-number* and make it accessible only by you:**

```
chmod 700 your-roll-number
```

  Then change to that directory and do your work.
- **vector operations are a must or lose lots of marks!!**
- **Label all plots. Add legends. Make the plots professional looking.**
- **Comments are not optional. They are required.**
- **pseudocode should be readable and neatly formatted.**
- **code should be written as part of a LyX file. The LyX file should be professional in appearance, and I will give marks for it.**
- **LyX file should be named *your-roll-number.lyx***
- **PDF file should be named *your-roll-number.pdf***
- **Python code should be named *your-roll-number.py***
- **Python code should run!!**
- **Include the plots in the lyx file and generate a pdf.**
- **Internet will be turned off at the beginning of the exam.**
- **Leave the above three files in your working directory**
- **Late submission will result in reduced marks.**

This is a problem in radiation from a loop antenna of length λ.

A long wire carries a current

$$I = \frac{4\pi}{\mu_0}\cos(\phi)\exp(j\omega t)$$

through a loop of wire. Here, ϕ is the angle in polar coordinates i.e., in (r,ϕ,z) coordinates. The wire is on the $x-y$ plane and centered at the origin. The radius of the loop is 10 cm and is also equal to $1/k = c/\omega$. The problem is to compute and plot the magnetic field $\vec{B}$ along the z axis from 1cm to 1000 cm, plot it and then fit the data to $\left|\vec{B}\right| = cz^b$.

The computation involves the calculation of the vector potential

$$\vec{A}(r,\phi,z) = \frac{\mu_0}{4\pi}\int\frac{I(\phi)e^{jkR}ad\phi}{R}$$

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

$$A_{ijkl} = \sum_{l=0}^{N-1}\frac{\cos(\phi_l')\exp(jkR_{ijkl})\,\vec{dl}'}{R_{ijkl}} \tag{1}$$

where $\vec{r}$ is at r_i, ϕ_j, z_k and $\vec{r}'$ is at $a\cos\phi_l'\hat{x} + a\sin\phi_l'\hat{y}$.

From $\vec{A}$, you can obtain $\vec{B}$ as

$$\vec{B} = \nabla\times\vec{A}$$

Along the z axis this becomes

$$B_z(z) = \frac{A_y(\Delta x,0,z) - A_x(0,\Delta y,z) - A_y(-\Delta x,0,z) + A_x(0,-\Delta y,z)}{2\Delta x + 2\Delta y} \tag{2}$$

Use $\Delta x = \Delta y = 1$cm.

- ▷ **Write pseudocode for how you will solve this problem.**
- ▷ **Break the volume into a 3 by 3 by 1000 mesh, with mesh points separated by 1 cm. The 3 by 3 grid in $x-y$ is to compute the curl using Eq. 2.**
- ▷ **Break the loop into 100 sections. Plot the current elements in $x-y$ (place points at the centre points of the elements. Properly label the graph.**
- ▷ **Obtain the vectors $\vec{r}_l'$, $\vec{dl}_l$ and $\vec{r}_{ijk}$, where l indexes the segments of the loop and i,j,k index volume of space where the vector potential is required.**
- ▷ **Define a function `calc(l)` that calculates and returns $\vec{R}_{ijkl} = \left|\vec{r}_{ijk} - \vec{r}_l'\right|$ for all $\vec{r}_{ijk}$ (l is the index into the $\vec{r}'$ array, which you have defined earlier.) Note: vectorize this**

▷ **Now compute $\vec{B}$ along the z axis. Use Eq. 2; remember to vectorize.**

▷ **Plot the magnetic field $B_z(z)$. Use a log-log plot.**

▷ **Fit the field to a fit of the type $B_z \approx cz^b$.**

▷ **Discuss your finding. Does B_z fall off as expected? What decay rate would you have expected for a static magnetic field? Where is the difference coming from?**

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=...)
```

```
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()
```