# EE2703 : Applied Programming Lab
# End Semester

Adityan C G EE19B003

May 2021

## Objective

The objective of this assignment is to analyse problem in radiation from a loop antenna whose current is given as,

$$I = 4\pi/\mu_o * cos(\phi) * exp(j * \omega * t)$$

We mainly analyse the field radiated by calculating it through vector operations.

## Theory

In the current equation $\mu_o$ is the Magnetic permeability of the medium, where $\phi$ is the polar angle from the polar coordinates $r, \phi, z$. It is given that the loop antenna is on the x-y plane and also centered at the origin.

Also the radius $r$ of the loop is equal to the wave number of the signal produced.

$$r = 2\pi/\lambda = 1/k = c/\omega \tag{1}$$

This also means that the circumference is the wavelength of signal. The problem at hand is to compute and plot the magnetic field on the z-axis from 1 to 1000 cm and then fit the data to the function, $\left|\vec{B}\right| = cz^b$. Also the main challenge in this exam is to handle Python arrays efficiently, and also to understand what accuracy a particular choice of grid will give.

Firstly the Magnetic Potential computation is done by,

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

where k is 1/r (r = radius) and $\vec{R} = \vec{r} - \vec{r'}$, where $\vec{r'} = r\hat{r'}$ is the point on the loop. Due to the This can be reduced to a sum:

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

where $\vec{r} = r_i$, $\phi_j$, $z_k$ and $\vec{r'} = \mathrm{r}\cos(\phi_l')\hat{x} + \mathrm{r}\sin(\phi_l')\hat{y}$. Also the above equation is valid for any $x_i$,$y_j$,$z_k$ in the space. This should be vectorized over 'l' in python. Now, from $\vec{A}$ we can obtain $\vec{B}$ by,

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

The equation can be approximated for z component of $\vec{B}$ i.e.$B_z(z)$ as,

$$B_z(z) = \frac{A_y(\Delta x, 0, z) - A_x(0, \Delta y, z) - A_y(-\Delta x, 0, z) + A_x(0, -\Delta y, z)}{4\Delta x \Delta y}$$

## Pseudo Code

- Defining angle $\phi$ such that it breaks the loop into 100 sections.
- Defining current elements and using quiver command to plot the current elements in the frame of loop sections
- Defining $\vec{r}$ from mesh grid X,Y,Z and $\vec{r'}$ from $\phi$ and $\vec{dl}$ from $\phi$ vector calculating Rijkl magnitude by $Rijkl = |\vec{r} - \vec{r'}|$
- Then looping over l to calculate $A_x$ and $A_y$,
- Now computing $B_z$ by,

  $B_z$ = ($A_y$[-1,1,:] - $A_x$[1,-1,:] - $A_y$[0,1,:] + $A_x$[1,0,:])/(4*1e-4)

## Meshgrid

As per the given mesh grid dimensions converting everything in SI units we can define it as, **Ideally on the z axis the z-component of Magnetic field is zero, so we are difining the grid just off zero for either x or y.**

```
x = np.linspace(-0.00999,0.01,3)
y = np.linspace(-0.01,0.01,3)
z = np.linspace(0.01,10,1000)
X,Y,Z = np.meshgrid(x,y,z,indexing = 'ij')
```

## Current plot

To break the loop into 100 sections we first define the angle $\phi$ as,

```
sec = 100
phi = np.array(np.linspace(0, 2*np.pi, sec, endpoint = False))
r_ = np.array([r*np.cos(phi),r*np.sin(phi)]).T
```

And for the current, the vector definition I.$\vec{dl}$ and the quiver plot is done by,

```
I = 4*np.pi/mu0*np.array([-np.sin(phi)*np.cos(phi),np.cos(phi)*np.cos(phi)]).T
plt.quiver(r_[:-1,0],r_[:-1,1],Ix,Iy)
```

where Ix and Iy place the current elements in the midpoints of the positions of $\vec{r'}$ and is given as,

```
Ix = 0.5*(I[0:-1,0] + I[1:,0])
Iy = 0.5*(I[0:-1,1] + I[1:,1])
```

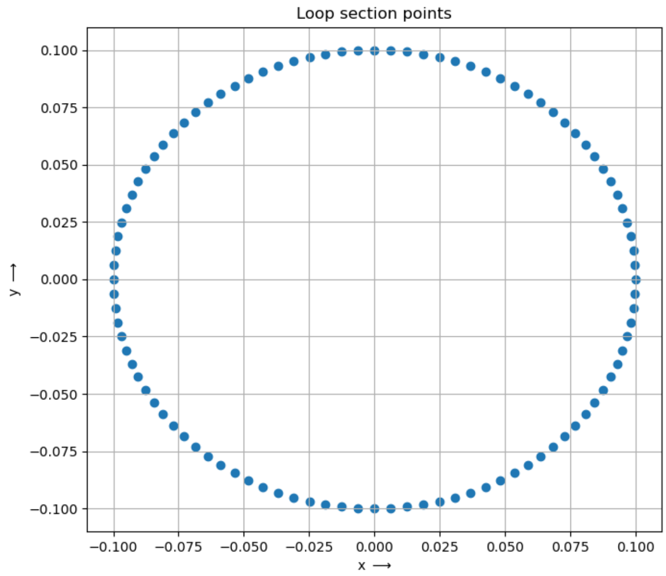


Figure 1: Sections in the loop

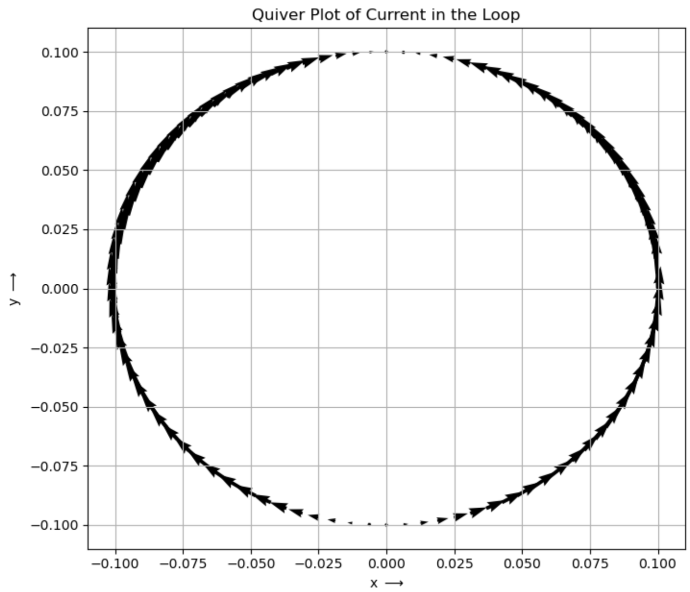


Figure 2: Current elements in the midpoint sections of the loop

## Vectors in the Loop

Define $\vec{r'}$ and $\vec{dl}$ as,

```
 r_ = np.array([r*np.cos(phi),r*np.sin(phi)]).T
dl = (2*np.pi*r/sec)*np.array([-np.sin(phi),np.cos(phi)]).T
```

because, dl vector is along $\hat{\phi}$

## Computation of Rijkl and Aijkl

The vectors of $\vec{r}$ are for each 'l'. There are 9000(3×3×1000) points in the mesh grid.So the vector can be obtained from the meshgrid as,

```
rvec = np.array((X,Y,Z))}
```

The Magnitudes $Rijkl$ for one value of l can be given by,

```
Rijkl = np.sqrt((rvec[0,:,:,:]-r_l[0])**2+(rvec[1,:,:,:]-r_l[1])**2+rvec[2,:,:,:]**2)
```

where $r_l$ is,

```
r_l = np.array((r_[l,0],r_[l,1],0))
```

Shape of the vector rvec containing all points: (3, 3, 3, 1000) contains all the x,y,z coordinates of 9000 grid points.For Aijkl, corresponding to a single value of 'l' is defined as,

```
Aijkl = np.cos(phi[l])*np.exp(-1j*k*Rijkl)/Rijkl
```

Now in a for loop we can either get Aijkl or Rijkl from the calc(l): function and compute Ax and Ay separately,

```
#if Rijkl is returned,
Ax = np.cos(phi[l])*dl[l][0]*np.exp(-1j*k*R)/R
A_x = A_x + Ax
Ay = np.cos(phi[l])*dl[l][1]*np.exp(-1j*k*R)/R
A_y = A_y + Ay

#if Aijkl is returned,
Aijkl = calc(l)
A_x = A_x + Aijkl*dl[0]
A_y = A_y + Aijkl*dl[1]
```

## Magnetic Field Computation and plot

From the equation given in the theory part for Bz, it is vectorized as,

```
B_z = (A_y[-1,1,:] - A_x[1,-1,:] - A_y[0,1,:] + A_x[1,0,:])/(4*1e-4)
```

The loglog plot can be plotted with respect to z as,

```
plt.loglog(z,np.abs(B_z),label = 'Bz(z)')
```

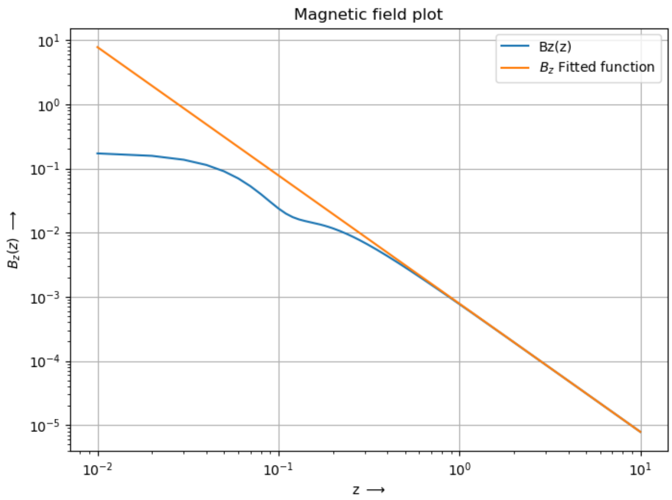


Figure 3: Magnetic Field plot and Fitted function

## Fitting the Magnetic field

As we can see from the magnetic field plot, the beginning points will not fit in the graph of $c * z^b$. So we take the graph from latter 700 points, and try to fit it with the Magnetic field values that we obtained.

$$B_z = c * z^b log(B_z) = log(c) + b * log(z)$$

In the matrix form, the equation will be, A*x = $B_z$ where x,A and $B_z$ are,

$$\mathrm{x} = \begin{pmatrix} log(c) \\ b \end{pmatrix} \ \mathrm{A} = \begin{pmatrix} 1 & z[0] \\ 1 & z[1] \\ . & . \\ . & . \\ 1 & z[999] \end{pmatrix} \ B_z = \begin{pmatrix} \mathrm{B}_z[0] \\ \mathrm{B}_z[1] \\ .. \\ .. \\ \mathrm{B}_z[999] \end{pmatrix}$$

By applying least squares, we get the fitted values for the field values above z = 300cm

```
log_c,b = np.linalg.lstsq(A,np.log(np.abs(B_z[300:])),rcond = -1)[0]
c = np.exp(log_c)
```

We can also add the fitted graph in the plot of magnetic field as shown in the figure above,

```
y = c*np.power(z,b)
plt.loglog(z,y,label= r'$B_z$ Fitted function')
```

and the fitted graph is also labelled along with the Field plot in Figure:3

## Analysis and Conclusion

From the plot of the Magnetic field we realize that the z-component of the magnetic field is decaying from 1e-1 to 1e-5. These are the z-component magnetic fields just off-axis, i.e.(0.009995,0,z)

**Changing the grid** :
Now let's see what happens if we define the grid exactly, such that we obtain the z-component values of B on z axis.

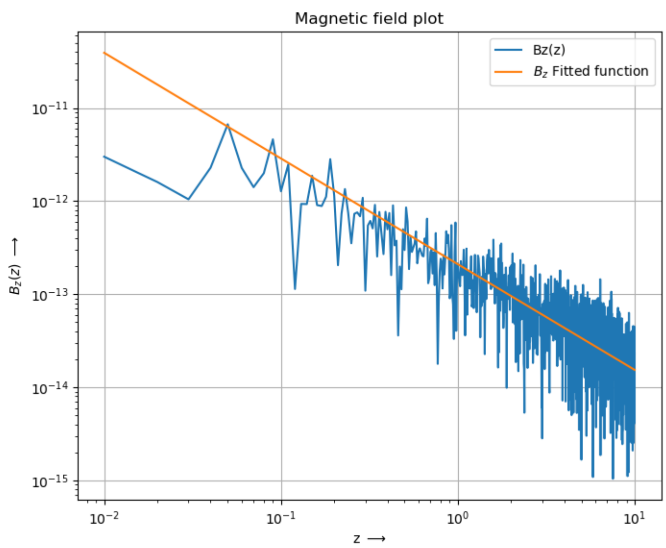


Figure 4: Sections in the loop

We can observe that the magnitude is in the order of 1e-12 and decaying i.e. almost zero. And also the error or noise is due to error in the differentiation of about 1cm $\delta x = \delta y = 1cm$. The magnitude ideally should be exactly zero due to the symmetric current function,

$$I = 4\pi/\mu_0 * cos(\phi)$$

And the fitting curve has linear decay, as we obtain the values of b and c as,

b = -1.135111454472268
c = 2.1022379916090177e-13

This is due to the 1/Rijkl term as the numerator terms in Ax and Ay add upto almost zero.

Or this can be inferred by:
Ideally without any errors, the $B_z$ is zero on the z axis, so if we take limit the 1/R terms remains without cancelling. So the dependency is -1 approximately in this case. This can also be obtained by applying limits.

This should be wrong. Because the values are ideally zero and z should not have any dependency i.e. $\mathbf{b = 0}$ , on the z axis.

**Observation** :

Not only just off axis but if we move across any other grid, example: x and y can be across anything independently

```
1.) x or y = np.linspace(0,2,3) #(0,1,2)
2.) x or y = np.linspace(9,11,3) #(9,10,11)
3.) x or y = np.linspace(85,87,3) #(85,86,87)
```

we get b values as,

$b_1$ = -1.9985854205495048
$b_2$ = -1.9976385584487188
$b_3$ = -1.9305691317669393

We get b = -2 approximately. By our common knowledge in physics, we know that the magnetic field $B$ along x varies as $\frac{z}{R^3}$, where $R$ is $\sqrt{z^2 + r^2}$, r is the radius of the loop. Thus, for large $z$, we see that $B$ along x varies as $z^{-2}$, so this shows that the non-zero z-components of the Magnetic field across the space due to this loop antenna varies with z as a dependency of **-2**.