EE2703 : Applied Programming Lab
-
Final Exam
-
Magnetic Field due to a current in loop

T.M.V.S GANESH
EE19B124

June 1, 2021

Introduction

The assignment is about

- Calculating the z component of magnetic field due to a current carrying wire in the x-y plane
- The variation of field along the z-axis at a constant x and y.

Description:

Current Equation:

A cuurent carrying circular wire is placed on the x-y plane with origin as the centre and radius of 10cm .
The current varies with the polar angle and time as

$$I(\phi) = \frac{4\pi}{\mu_0}\cos(\phi)e^{j\omega t} \tag{1}$$

The time variance of the current is not considered during the calculation and is calculated at time t = 0.

Where ϕ is the polar coordinate in (r,ϕ,z)

Vector Potential

The Vector potential due to the current is given by:

$$\vec{A}(r,\phi,z) = \frac{\mu_0}{4\pi}\int \frac{I(\phi)\vec{\phi}e^{-jkR}ad\phi}{R} \tag{2}$$

where R = $|\vec{r} - \vec{r_l}|$, $\vec{r}$ is the point at which potential is being calculated and $\vec{r_l}$ is the point on the wire at a particular ϕ.
The integral can be written as a sum of values sigma at various ϕ values divided into N equal sections in the xy plane.
Thus A(potential) at a certain (i,j,k) point can be calculated as:

$$\vec{A_i}jk = \sum_{l=0}^{N-1} \frac{cos(\phi_l)e^{-jkR_{ijkl}}\vec{dl}}{R_{ijkl}} \tag{3}$$

Magnetic Field thus can be calculated from the potential vector as :

$$\vec{B} = \nabla X \vec{A} \tag{4}$$

As the magnetic potential is along $\vec{\phi}$ the x and y components of the magnetic field goes to zero and z-component can be calculated from the A vector as:

$$B_z(z) = \frac{A_y(\Delta x, 0, z) - A_x(0, \Delta y, z) - A_y(-\Delta x, 0, z) + A_x(0, -\Delta y, z)}{4\Delta x \Delta y} \tag{5}$$

Q1 - Pseudo Code:

1.Divide the volume into a 3 by 3 by 1000 mesh with points separated by 1cm.

- The use of meshgrid function creates a $3 \times 3 \times 1000$ Matrix of X,Y,Z coordinates and can be used to compute the Value of R

2.Break the loop in the X-Y axis into N sections so that the integration can be approximated as a sum of values at these points.

- The value of ϕ can be linearly divided between 0 and 2π

- Radius vector($\vec{r_l}$) and the tangential vector($\vec{dl}$) the direction of the current can be calculate from ϕ vector.
- Thus Current vector can be computed from $\vec{dl}$ and plotted

3.Define a function calc(l) which then used to calculate R and the x and y component of potential at a particular l.

- These can be added over all values of l to obtain the x and y components of the magnetic potential.

4.The Magnetic Field is calculated using the x and y components of potential from the equation above.

- plot Magnetic Field vs z using loglog plot.

5.Fit the data to a exponential using Least Squares approach.

Q2 Meshgrid

```
#Q2 - Divinding the Volume into a 3 X 3 X 1000 Meshpoints

i = np.array([0,1,2])
            #X-Coordinates
j = np.array([0,1,2])
            #Y-Coordinates
k = np.linspace(1,1000,1000)
            #Z-Coordinates from  1 to 1000cm with a step of 1cm.
xx,yy,zz= meshgrid(i,j,k)
            #Meshgrid
```

xx,yy,zz denote meshpoints of the respective i,j,k coordinates.

since the field is symmetrical and net sum of the current adds to zero.The magnetic field along the z-axis is zero thus I have considered a different axis that is x=1,y=1,z from 1 to 1000 i.e the axis in the equations are shifted.

Q3 - Dividing loop

```
#Q3- Dividing the current carrying loop into N Sections

r = 10
            #Radius of loop(r=a=k)
N = 100
            #No.of Sections
phi = np.linspace(0,2*np.pi,N)
            #Ploar angle array at the corresponding Section
x = np.array(r*(np.cos(phi)))
            #X-Coordinates of loop(r*cos(phi))
y = np.array(r*(np.sin(phi)))
            #Y-Coordinates of loop(r*sin(phi))
```

phi is array of the angles of the N sections of the loop and x,y denotes the X,Y coordinates of the sections of loop at particular ϕ

Q5 - $\vec{r_l}$ and $\vec{dl}$

```
#Q5- r_l and dl vectors

```

```
r_l = c_[x,y,np.zeros(N)]
            #3D vector positions of the sections in loop
dl  = c_[-y,x,np.zeros(N)]
            #Tangential Vector at Sections of loop(-rsin(phi)i+rcos(phi)j^)
```

$\vec{r_l}$ is the radius vector i.e the vector connecting origin and the section point for a particular ϕ and $\vec{dl}$ is the tangential direction i.e the direction of current flow in the loop.
r_l and dl are the arrays(column wise) representing the x,y,z components of these vectors.

Q4 - Current in the loop

```
#Q4- plot of Current Flowing in the Loop

u = 4*np.pi*(10**-7)
            #permeability of free space
I_x = ((4*np.pi)/u)*np.cos(phi)*dl.T[0]
            #X - component of Current in the loop
I_y = ((4*np.pi)/u)*np.cos(phi)*dl.T[1]
            #Y - component of Current in the loop

figure("Current in the loop",figsize = (8,8))
            #Figure
quiver(x, y, I_x, I_y,headlength=5,headwidth = 5,width=0.003,pivot = 'mid')
            #Quiver plot of Current in loop
#colorbar()
xlabel("X-coordinates",fontsize = 15)
ylabel("Y-Cordinates",fontsize = 15)
title("Current Vector in the loop",fontsize = 15)
xticks(np.arange(-10, 11, step=1))
yticks(np.arange(-10, 11, step=1))
grid()
```

I_x and I_y represent the x and y components of current.The z-component of the current is zero.

Plot:

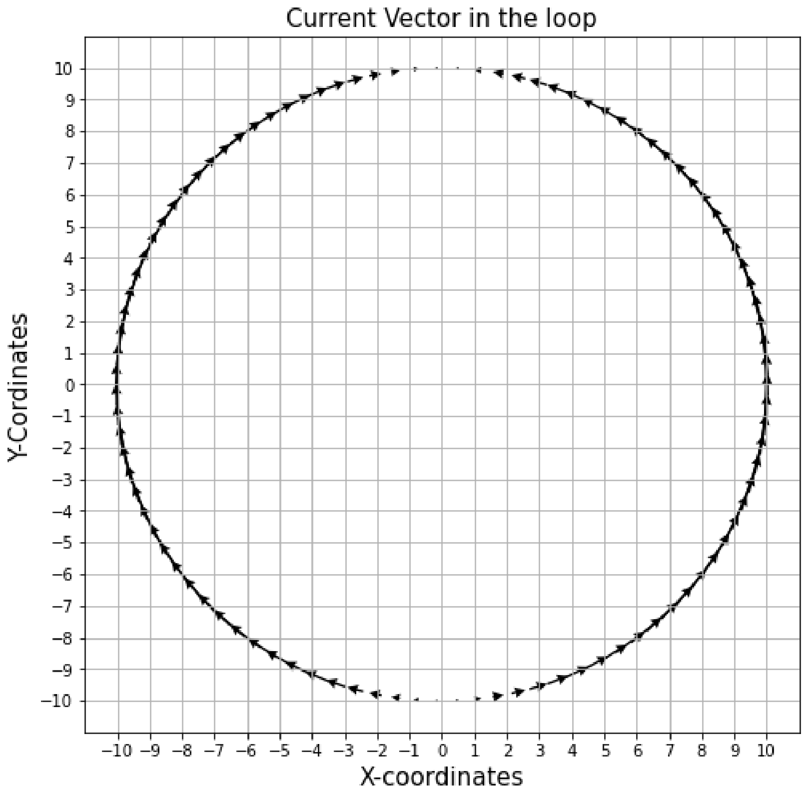

Figure 1: Quiver plot of current in the loop

Q6,7 - calc function and calculating Potential Vector

```
#Q6,7 Defining the calc function to calculate R_ijkl and extending it to find A
    vector

def calc(l):
            #function to calculate R,A matrices
    R_ijkl = np.sqrt(((xx-r_l[l][0])**2)+((yy-r_l[l][1])**2)+(zz**2))
            #The value of R_ijkl all over the 3X3X1000 meshgrid for a particular l
    A_ijkl_x = (np.cos(phi[l])*(dl[l][0]) * (1/R_ijkl))* (np.exp(-1j*0.1*R_ijkl))
            #The x component of potential all over mesh grid for a particular l
    A_ijkl_y = (np.cos(phi[l])*(dl[l][1]) * (1/R_ijkl))* (np.exp(-1j*0.1*R_ijkl))
            #The y component of potential all over mesh grid for a particular l
    return A_ijkl_x,A_ijkl_y,R_ijkl

def potential():
            #Function to calculate the potential Vector
    A_x = calc(0)[0]
            #Initialing A_x 3X3X1000 matrix with l= 0 from calc fn
    A_y = calc(0)[1]
            #Initialing A_y 3X3X1000 matrix with l= 0 from calc fn
```

```
    for l in range(1,N):
        A_x += calc(l)[0]
            #X component of potential for all (i,j,l)
        A_y += calc(l)[1]
            #X component of potential for all (i,j,l)
    return A_x,A_y

A_x,A_y= potential()
```

The calc function returns the value of R and the term to get added for potential calculation at a particular l all over the meshgrid and potential() function uses for loop to add these terms for all ϕ of sections to get the Magnetic Potential.

Q8 - Computing B_z

From the above equation 5,We can consider that $\Delta x = \Delta y = 1cm$ and shift the values by 1cm since we are considering the axis as (x=1,y=1) instead of z-axis,The indices in the potential don't change with the shifting of axis

Vectorised Code:

```
#Q8 - Calculating Z component of Magnetic field

B_z = (A_y[2,1,:]-A_x[1,2,:]-A_y[0,1,:]+A_x[1,0,:])/(4)
            #Magnetic field according to equation 2 in the pdf
```

This creates a array of size 1000 and contains the values of z component of Magnetic field at corresponding value of z from 1 to 1000cm.

Q9 - Plotting the Magnetic Field

Code:

```
#Q9 - loglog plot of the Magnetic field

figure("Magnetic Field ",figsize = (8,6))
loglog(k, np.abs(B_z),label = "B_z")
            #log(B) vs log(z) plot of magnetic field
xlabel("Z-Cordinates(log)",fontsize = 15)
ylabel("Z comp of Magnetic field(B_z)",fontsize = 15)
title("Magnetic field along x=1,y=1,z",fontsize = 15)
```

```
legend(loc = 'upper right',fontsize = 15)
grid()
#show()
```

Log log plot of the Magnetic Field vs z :
Plot:

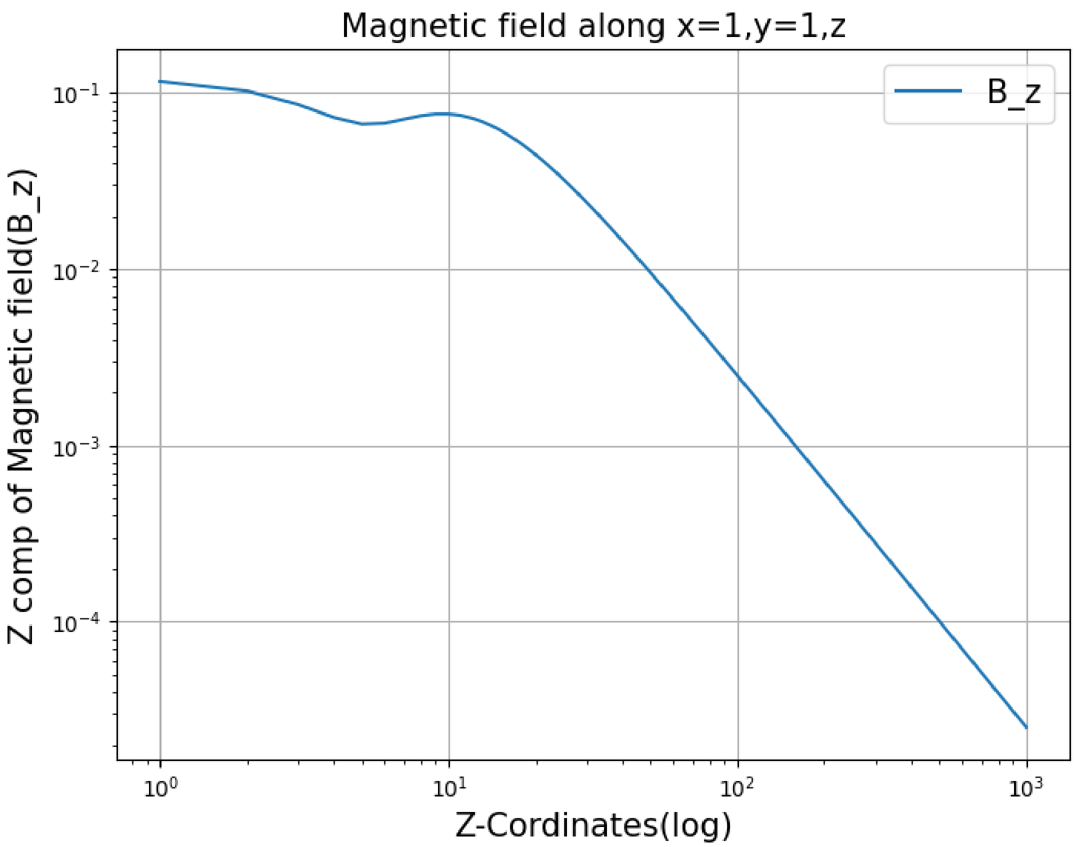

Figure 2: Loglog plot of $B_z vsz$

Q10 - Fitting data to a exponential

Since the loglog plot appears to be a straight line for most part of the graph we try to fit the data to a exponential $B_z = c*(z^b)$ using the least square approach. Code:

```
#Q10 - Fitting the data to a exponential using least squares method

M = c_[np.ones(1000),np.log(k)]
        #M*p = X ,M is a vector of 1, log(k)
X = c_[np.log(np.abs(B_z))]
        # X is vector of log(Magnetic field)
p = lstsq(M,X,rcond=None)
```

```
c = np.exp(p[0][0])
b = (p[0][1])
print("The Value of c and b are:")
print(c,b)
B_fit = c*(k**b)
            #value of fitted data
figure("least square Fit",figsize = (8,6))
            #Graph to compare both the plots
loglog(k, np.abs(B_z),label = "Original")
loglog(k, np.abs(B_fit),label = "fit")
xlabel("Z-Cordinates(log)",fontsize = 15)
ylabel("Z comp of Magnetic field(B_z)",fontsize = 15)
title("Magnetic field fit vs original",fontsize = 15)
legend(loc = 'upper right',fontsize = 15)
grid()
show()
```

Result:

```
The Value of c and b are:
[9.01341317] [-1.83256729]
```

Plot:

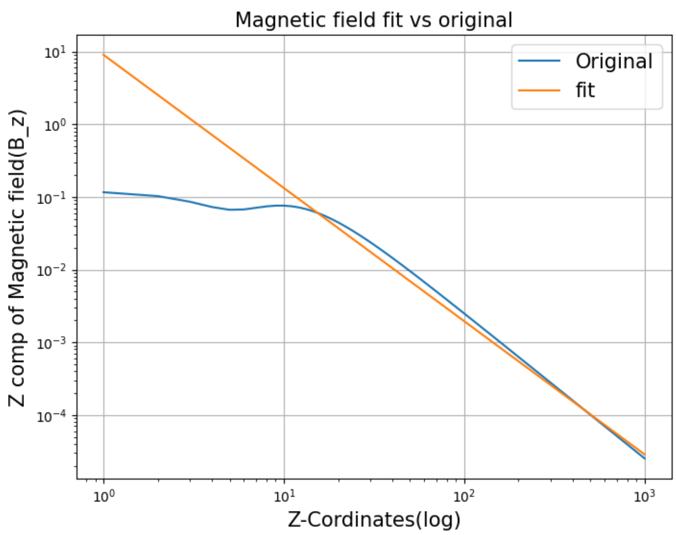

Figure 3: Loglog plot of B_z and the obtained exponential

Q11 - Conclusion

1.The usage of arrays increases the efficiency of the code as it replaces the nested for loops and execution actually is done in C language.

2.We observe that the magnetic field is nearly zero along the z-axis due to symmetry of the current and one cancels out the effect of another.

3.For a different axis i.e along x=1,y=1 in this case the magnetic field remains constant upto a certain distance and decays exponentially as z increases.

4.For a spatially non varying current we expect the decay rate to be -3 for large values of z and is approximately constant for small values of z.

5.In this case we observe that it is constant for smaller values of z but the decay rate is nearly -2 instead of the expected -3.

6.The difference in the decay rate is due to cos(ϕ) term in the current expression.