EE2703: Applied Programming Lab
End Semester

Devaganthan S S
EE19B018

August 6, 2021

0.1 Abstract

This experiment aims to compute and plot the magnetic field $\overrightarrow{B}$ along the z axis from 1cm to 1000cm, and fit the data to the model $B_z = c.z^b$.

0.2 Introduction

The first step in finding the magnetic field is to compute the vector potential. This can be done with the below equation,

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

Where $\overrightarrow{R} = \vec{r} - \overrightarrow{r}'$ and $k = \omega/c = 0.1$. $\overrightarrow{r}$ is the point where we want the Field, and $\overrightarrow{r'} = a\hat{r'}$ is the point on the loop.

Then we compute the curl of the vector potential to find the magnetic field. Since the assignment requires only the component along the Z-axis, we employ the below equation,

$$B_z(z) = \frac{A_y(\Delta x, 0, z) - A_y(-\Delta x, 0, z) - A_x(0, \Delta y, z) + A_x(0, -\Delta y, z) -}{4\Delta x \Delta y} \tag{2}$$

We then fit the data to the equation $|B_z| = c.Z^b$ using the least-squares method.

0.3 Pseudo Code

1. We break the volume into 3 x 3 x 1000 mesh. This Mesh represents the $\vec{r}_{ijk}$
2. We break the Loop of wire into 100 sections. The section coordinates are of the form [10cos(θ),10sin(θ)] where θ goes from 0 to 2π (100 divisions). The direction of current is tangential to the section. So we multiply the coordinates with the direction and plot the quiver plot.
3. Using the Calc(l) function we generate the terms in the summation (Vector Potential Equation). We add the terms in a 'for loop'.
4. In calc(l),
    - We first Compute the $\vec{R_{ijk}}$. This done by subtracting the $r_{\vec{ijk}}$ mesh from $r_{\vec{ijk}}'$. (NumPy Subtraction)

- Then Magnitude of $\vec{R_{ijk}}$ is computed and with this the vector potential is computed due to l^{th} section of the for loop.

5. After computing the Vector Potential, using Equation 2 we can compute the B_z. This is done with simple NumPy addition and subtraction of arrays.

6. To fit the data to the model, we first make the appropriate NumPy Arrays. Then using scipy.linalg we find the necessary parameters.

0.4 Implementation and Results

0.4.1 Breaking the volume into a mesh grid (Q2)

With the function **meshgrid()**, we create a 3 x 3 x 1000 mesh, with a 1 cm gap between two adjacent mesh points. The below code accomplishes the above tasks.

```
# Question 2 - Breaking the Volume into 3 by 3 by 1000 Mesh
x = arange(-1, 2, 1)
y = arange(-1, 2, 1)
z = arange(1, 1001, 1)
xx, yy, zz = meshgrid(x, y, z)  # Components of rijk
```

0.4.2 Plotting the Current Elements in XY plane (Q3)

The magnitude of the Current is calculated first at each section of the wire. The direction of the current is tangential to the loop. Using matplotlib, the quiver plot is plotted. The below code accomplishes the above task,

```
# Question 3 - Current Elements in XY Plane
mu0 = 4*pi*1e-7
theta = linspace(0, 2*pi, 100)  # Dividing the loop into 100 sections
I = (4*pi)/mu0*cos(theta)  # Mag of I at each section of the loop
rx = 10*cos(theta)
ry = 10*sin(theta)
dirX = I*-1*sin(theta)  # Direction of the Current in x Direction
dirY = I*cos(theta)  # Direction of the Current in y Direction

fig, ax = plt.subplots()  # Quiver plot for the current
ax.quiver(rx, ry, dirX, dirY, color='black')
grid()
xlabel('X axis')
ylabel('Y axis')
title('Current Elements of the Loop')
show()
```

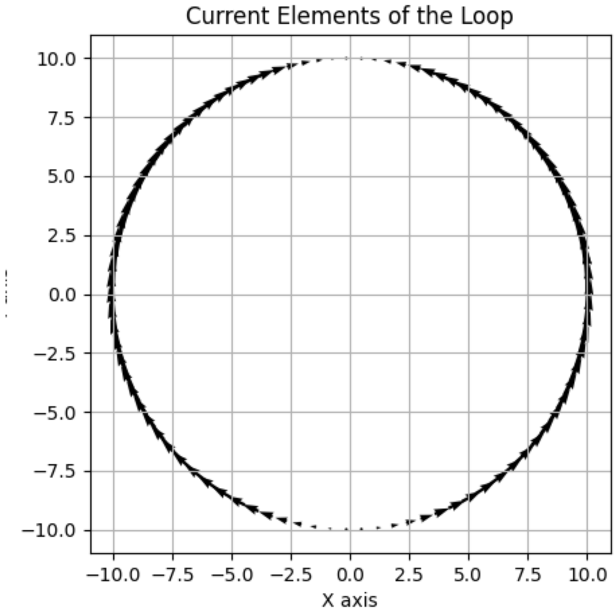

Figure 1

0.4.3 Obtaining the Vectors $\vec{r_l'}$ and $\vec{dl'}$ (Q4)

The x and y components of the sections are 10cos(θ) and 10sin(θ). The $\vec{dl}'$ is the tangential vector to the loop at each section. The below code computes the above vector.

```
# Question 4 -  The Vectors rl' and dl
r1ijk = [rx, ry, zeros(1000)]  # Rx and Ry are already computed
dlijk = [-1*sin(theta), cos(theta), zeros(1000)]  # Tangential Direction
```

0.4.4 Defining the Function calc(l) and extending it to compute the Vector Potential (Q 5 , Q6)

Firstly, we compute the R_{ijk} vector and find the magnitude of it at each point on the mesh. Then the function calc(l) computes,

$$calc(l) = \frac{cos(\phi_l')exp(-jkR_{ijkl})\overrightarrow{dl'}}{R_{ijkl}} \tag{3}$$

Below Code is the Function calc(l),

```
# Question 5 and 6 - Calc(l) function
def calc(l):
    Rx = xx-ones(1000)*10*cos(theta[l])  # X component of Rijk
    Ry = yy-ones(1000)*10*sin(theta[l])  # Y compoennt of Rijk
    Rl = (Rx**2+Ry**2+zz**2)**0.5  # Mod of Rijk
    Ax = (cos(theta[l])*numpy.exp(-0.1j*Rl)*-1*sin(theta[l]))/Rl
    Ay = (cos(theta[l])*numpy.exp(-0.1j*Rl)*cos(theta[l]))/Rl
    return Ax, Ay
```

0.4.5 Computing the Vector Potential (Q7)

Using the function calc(l), A_{ijk} is computed in a for loop. Since the number of iterations are 100 and we are adding NumPy arrays, the process is bearable. The trade off is, easier implementation. The below code computes the vector

```
# Question 7 - Calculating the Vector Potential
A_x = 0  # Intializing
A_y = 0
for i in range(100):
    Ax, Ay = calc(i)
    A_x += Ax
    A_y += Ay
```

0.4.6 Computing $\vec{B}$ and Plotting it (Q8 and Q9)

We compute the Bz component of Magnetic Field using Equation 2.

```
# Question 8 - Computing the Magnetic Field along z axis
B = (A_y[1, 2, :] - A_x[1, 1, :] - A_y[1, 0, :] + A_x[2, 1, :])/4

# Question 9 -  Plotting the Magnetic Field
loglog(z, abs(B))
grid()
xlabel(’Z’)
ylabel(’|B(z)|’)
title(’Magnitude of Z component of Magnetic Field’)
show()
```

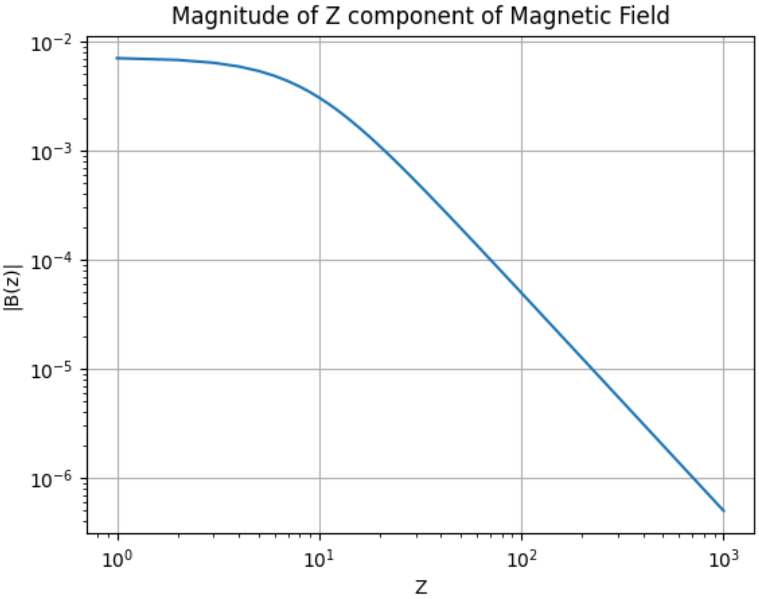

Figure 2

0.4.7 Fitting the Field to a Model Q10

Using least squares method, we try fitting the data – B_z into the model , $B_z = cZ^b$.

$$\begin{bmatrix}1 & log(z)\end{bmatrix}\begin{bmatrix}A\\B\end{bmatrix} = \begin{bmatrix}log(B_z)\end{bmatrix} \ (4)$$

Here, $c = exp(A)$ and $b = B$

```
# Question 5 and 6 - Calc(l) function
def calc(l):
    Rx = xx-ones(1000)*10*cos(theta[l])  # X component of Rijk
    Ry = yy-ones(1000)*10*sin(theta[l])  # Y compoennt of Rijk
    Rl = (Rx**2+Ry**2+zz**2)**0.5  # Mod of Rijk
    Ax = (cos(theta[l])*numpy.exp(-0.1j*Rl)*-1*sin(theta[l]))/Rl
    Ay = (cos(theta[l])*numpy.exp(-0.1j*Rl)*cos(theta[l]))/Rl
    return Ax, Ay
```

The Decay Rate Found is -1.9058964119106183. The fitted Plot is,

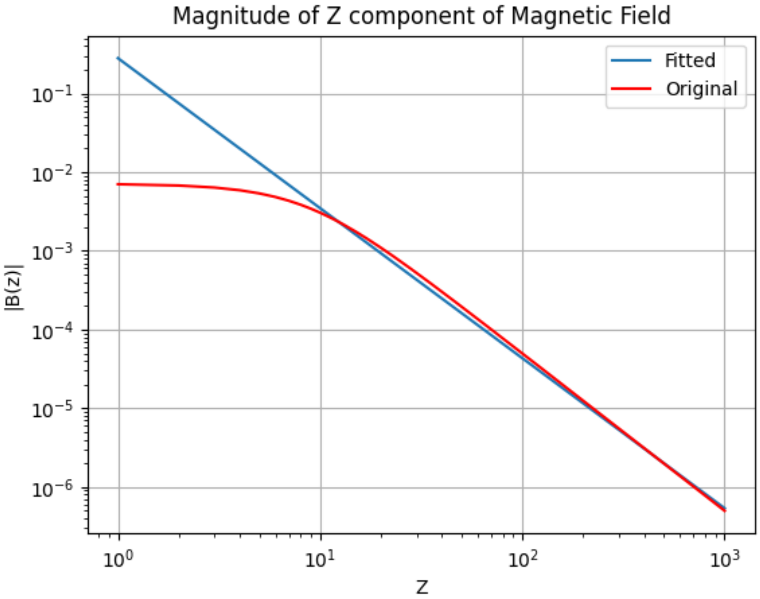

Figure 3

0.5 Conclusion Q11

The fitted B_z doesn't fall of as we expected. We find that the decay value as -1.9058964119106183. Ideally for a static Magnetic field the decay rate is -2. This difference is due to the grid we had chosen. The distance between two mesh points is 1cm, i.e ΔX = 1cm or ΔY = 1cm. The ΔX we choose, makes a significant effect on B for small values of z. But as z » x , y the effect is very negligible. Therefore, to get an accurate fit, we have to choose very small ΔX and ΔY