EE2703 - Endsem

EE19B094

May 2021

Abstract

We calculate the magnetic field for various points on the Z - axis, for a current carrying wire loop on the X-Y plane centered at origin of radius 10 units. We shall consider two cases i.e. one where current is static and one where current is changing with time (dynamic).

1 Q1 : Pseudo Code

1.1 Create a meshgrid

Make a meshgrid corresponding to the (3,3,1000) volume. Create r and r' matrix by copying r matrix 100 times.

1.2 r'_l and dl_l matrices

Make phi linspace by dividing (0,2π) into N parts, where N are the number of current elements on the wire loop
For the r_l matrix we convert the elements from spherical coordinates to cartesian coordinates.
For dl_l vectors we find them using $d\phi * radius$

1.3 R vector and Magnetic Potential

We calculate R by building a function *extended_calc()* which calculates $|r' - r_l|$ for all points.
Using these we calculate the magnetic potential A_x and A_y by summing dA over all the points l

1.4 Magnetic Field

We then shall take the curl of A, to get the magnetic field.

2 Q2 : Creating meshgrid

We create a mesh grid of the (3,3,1000) volume.

```
#Creating grid of the (3,3,1000) volume
x_points=np.linspace(-1,1,3)
y_points=np.linspace(-1,1,3)
z_points=np.linspace(1,1000,1000)
xx,yy,zz = np.meshgrid(x_points,y_points,z_points,indexing='ij')
meshgrid = np.stack((xx,yy,zz),axis=3)
position_vectors = np.tile(meshgrid,100).reshape(3,3,1000,100,3)
    #Position vector of all points
```

3 Q3 : Current elements and current

We first convert the position vectors of current elements from spherical to cartesian coordinates and plot them.
NOTE that we have already calculated r_l' vector here.

```
#Converting positions of current elements from (r,phi,z) to (x,y,z)
phi_vector = linspace(0,2*np.pi,N)
x_cap = radius*cos(phi_vector)
y_cap = radius*sin(phi_vector)
z_cap = zeros(N)
r_l = column_stack((x_cap , y_cap , z_cap))

#Plot current elements
title("Current elements on the conductor")
scatter(r_l[:,0],r_l[:,1])
ylabel(r"$Y\longrightarrow$")
xlabel(r"$X\longrightarrow$")
show()
close()
```

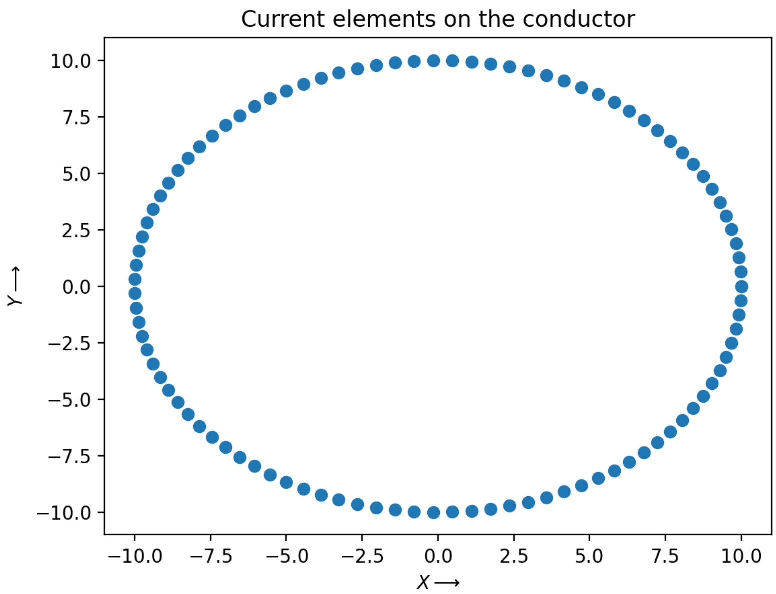

Figure 1: Current Elements

And now, we find the current and plot them using the quiver function for all the current elements.

```
#Calculate current
mu = 4*np.pi*pow(10,-7)

I_x = 4*np.pi/mu*(-cos(phi_vector)*sin(phi_vector))
I_y = 4*np.pi/mu*cos(phi_vector)*cos(phi_vector)
I_z = zeros(N)
I = column_stack((I_x , I_y , I_z))

#Quiver plot for I vs r
title("Quiver plot of current")
quiver(r_l[:,0],r_l[:,1],I[:,0],I[:,1])
ylabel(r"$Y\longrightarrow$")
xlabel(r"$X\longrightarrow$")
grid()
show()
close()
```

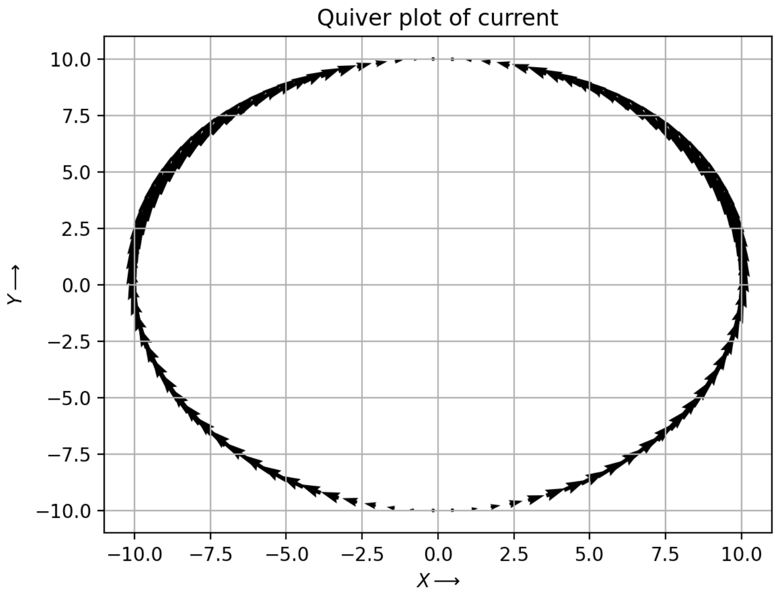

Figure 2: Quiver Plot of current

4 Q4 : r_l' and dl_l vectors

We find r_l' by converting the spherical to cartesian coordinates. We have already calculated it in the previous section to plot the current elements. While for dl_l, we calculate it using $d\phi * radius$

```
#Calculating dl
dphi_vector =
    column_stack((-sin(phi_vector),cos(phi_vector),zeros(N)))*(2*np.pi/N)
dl_vector = dphi_vector*radius
```

5 Q5,6 : calc() and extended_calc() function

We first define *calc()* function which gives the norm of the vector r-rl at any specified value of l.
We then extend this function to *extended_calc()* which gives the norm of the vector r'-rl for all values i.e. the R vector

```
#Function calc(l) to calculate norm of r-rl for a single point 'l'
def calc(r,r_l,l):
   return norm(r-r_l[l],axis=-1)

#Extended vectorized function of calc(l) to calculate R for all
    points
def extended_calc(r,r_l):
   return norm(r-r_l,axis=-1)

R_vector = extended_calc(position_vectors,r_l)
```

6 Q7 : Compute A vector

To find A we sum dA over all l. We find A for two cases i.e. when the current in the loop is dynamic i.e. time varying and second when it is constant. The integral form of magnetic potential is,

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

In our case, since the current is given by : $I = \frac{4\pi}{\mu_0}\cos(\phi)\exp(j\omega t)$ so the the integral can be reduced to the following summation:

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

```
#Function that returns value of the Green's function of R used in
    calculating Magnetic Vector Potential
def Green(R):
   return (e**(-1j*k*R))/R

dphi_magnitude = 2*np.pi/N
k = 0.1

#Calculating A,Bz for dynamic case
A_x_dynamic =
    np.sum((mu/(4*np.pi))*I[:,0]*Green(R_vector)*dphi_magnitude*radius,axis=-1)
A_y_dynamic =
    np.sum((mu/(4*np.pi))*I[:,1]*Green(R_vector)*dphi_magnitude*radius,axis=-1)

#Calculating A,Bz for static case
A_x_static =
    np.sum((mu/(4*np.pi))*I[:,0]*dphi_magnitude*radius/R_vector,axis=-1)
A_y_static =
    np.sum((mu/(4*np.pi))*I[:,1]*dphi_magnitude*radius/R_vector,axis=-1)
```

7 Q8 : Calculate magnetic field

We find the curl of the magnetic potential to calculate the magnetic field. Since,

$$B_z = \frac{dA_y}{dx} - \frac{dA_x}{dy} \tag{2}$$

we shall calculate it using,

$$\frac{Ay[2,1,:] - Ay[0,1,:])}{dx} - \frac{Ax[1,2,:] - Ax[1,0,:])}{dy}$$

to get value of Bz.

```
#Function to get Bz from A using curl
def get_Bz(Ay,Ax):
   #Taking Bz as the right side derivative of A
   Bz = (Ay[2,1,:]-Ay[0,1,:]-(Ax[1,2,:]-Ax[1,0,:]))
   return Bz

#Bz using right side derivative
Bz_dyanmic = get_Bz(A_y_dynamic,A_x_dynamic)

Bz_static = get_Bz(A_y_static,A_x_static)
```

8 Q10 : Fit magnetic field

We fit the magnetic field to the form cz^b and calculate the values of b and c for static and dynamic fields using *lstsq()* function.

However note that as Bz follows a linear graph in log scale only after a certain point, we ignore first *ignore_points* points for a better fit

```
#Function to fit the magnetic filed values on exponential curve
    using lstsq
#As Bz follows a linear graph in log scale only after a certain
    point, we ignore first 'ignore_points' points
#for a better fit
def exponential_fit(Bz,z,ignore_points):

   function_value = log(abs(Bz[ignore_points:]))
   x = log(z[ignore_points:])
   coefficients = column_stack((x,ones(1000)[ignore_points:]))
   b,c = lstsq(coefficients,function_value,rcond=None)[0]
   return b,c

#Fitting the values of Bz on bz^c
ignore_points = 15
b_dyanmic,c_dynamic =
    exponential_fit(Bz_dyanmic,z_points,ignore_points)
print("The dynamic magnetic field when fitted to cz^b gives
    log(c)=%f and b=%f"%(c_dynamic,b_dyanmic))

b_static,c_static =
    exponential_fit(Bz_static,z_points,ignore_points)
print("The static magnetic field when fitted to cz^b gives
    log(c)=%f and b=%f"%(c_static,b_static))
```

9 Q9 : Plot magnetic field

We plot the calculated and fitted values of magnetic field on loglog scale.

```
#Plotting absolute value of Bz vs z in loglog scale
x_range = log(z_points[ignore_points:])
title("Absolute value of Calculated and fitted values of Bz for
    dynamic case")
ylabel(r"log(|Bz|)$\longrightarrow$")
xlabel(r"log(z)$\longrightarrow$")
loglog(z_points,abs(Bz_dyanmic), label="Calculated values")
loglog(exp(x_range),exp(b_dyanmic*x_range+c_dynamic),label="Fitted
    values")
legend()
```

```
grid()
show()
close()

#Plotting absolute value of Bz vs z in loglog scale
x_range = log(z_points[ignore_points:])
title("Absolute value of Calculated and fitted values of Bz for
    static case")
ylabel(r"log(|Bz|)$\longrightarrow$")
xlabel(r"log(z)$\longrightarrow$")
loglog(z_points,abs(Bz_static), label="Calculated values")
loglog(exp(x_range),exp(b_static*x_range+c_static),label="Fitted
    values")
legend()
grid()
show()
close()
```

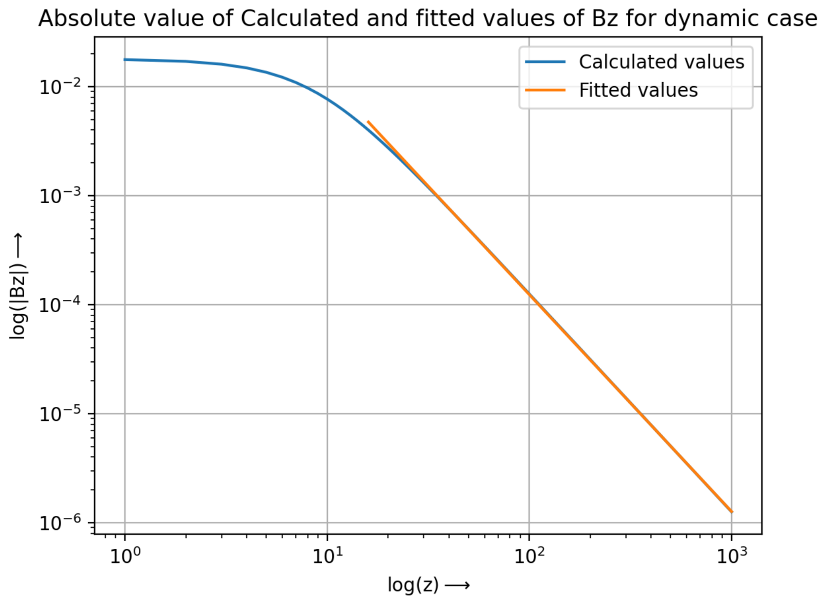

Figure 3: Field for dynamic current

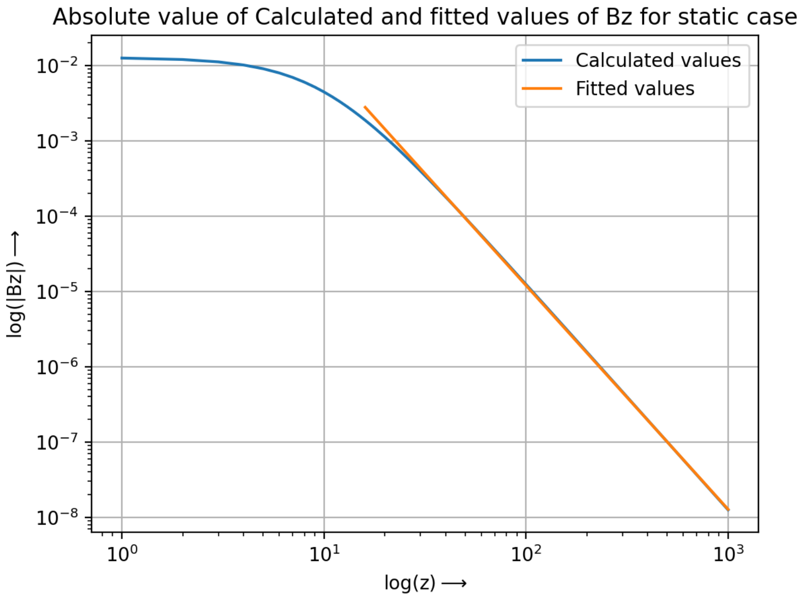

Figure 4: Field for static current

10 Q11 : Expected fall in magnetic field & Observations

The dynamic magnetic field when fitted to cz^b gives log(c)=0.160286 and b=-1.989159
The static magnetic field when fitted to cz^b gives log(c)=2.350873 and b=-2.971375

Thus we observe that, the magnetic field along Z-axis is proportional to z^{-2} as expected for a time dependant current. For a static current, the Magnetic Field is proportional to z^{-3}.

The main difference in both the cases is the term corresponding to the Green's function.

From, the maxwell's equation we see that the magnetic field might also depend on the Electric field. Now in the dynamic case, since E depends on inverse square of distance, that value dominates the inverse cube relation due to the current flow and hence making B proportional to z^{-2} (as compared to z^{-3} for large z). While in dynamic case, since $\frac{dE}{dt}$ is 0 in static case, B depends solely on current and is hence proportional to z^{-3} as expected.