EE2703 : Applied Programming Lab
End-Sem
Magnetic Field due to a Current loop

Krishna Somasundaram
EE19B147

May 30, 2021

Introduction

Our goal will be to evaluate the magnetic field along the z-axis for a current carrying loop placed in the x-y plane. We will first model the 3D space, the loop itself and find the magnetic field by evaluating the curl of the magnetic potential. Finally, we will attempt curve fitting in order to find the decay constant.

Theory

Firstly the Magnetic Potential computation is done by,

$$\vec{A}(r,\phi,z) = \frac{\mu_0}{4\pi}\int \frac{I(\phi)\hat{\phi}e^{(-jkR_{ijkl})}ad\phi}{R} \tag{1}$$

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. This can be reduced to a sum:

$$\vec{A}_{ijk} = \sum_{l=0}^{N-1} \frac{\cos{(\phi')}\, e^{(-jkR_{ijkl})}\overrightarrow{dl}'}{R_{ijkl}} \tag{2}$$

where $\vec{r}$ is at r_i, ϕ_j, z_k and $\vec{r'}$ is at rcos(ϕ_l')$\hat{x}$+ rsin(ϕ_l')$\hat{y}$. Now, from $\vec{A}$ we can obtain $\vec{B}$ by,

$$\vec{B} = \vec{\nabla} \times \vec{A} \tag{3}$$

Here, I have derived the formula as shown below. It does not match with what has been suggested in the Problem Statement. I have use the following formula to solve the problem instead.

$$\vec{B} = \nabla \times \vec{A} = \left(\frac{\partial Az}{\partial y} - \frac{\partial Ay}{\partial z}\right)\hat{x} + \left(\frac{\partial Ax}{\partial z} - \frac{\partial Az}{\partial x}\right)\hat{y} + \left(\frac{\partial Ay}{\partial x} - \frac{\partial Ax}{\partial y}\right)\hat{z} \tag{4}$$

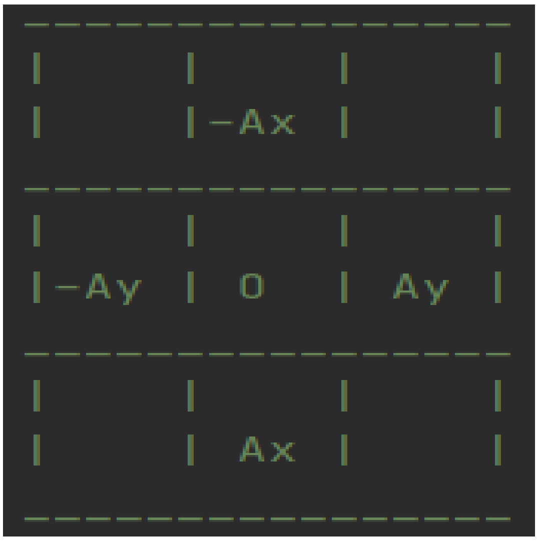
$$Bz = \left(\frac{\partial Ay}{\partial x} - \frac{\partial Ax}{\partial y}\right) \Longrightarrow \frac{Ay(\Delta x,0,z) - Ay(-\Delta x,0,z)}{2\Delta x} - \frac{Ax(0,\Delta y,z) - A(0,-\Delta y,z)}{2\Delta y} \tag{5}$$

We use vectorized code as opposed to a for loops to increase the speed of the program. The purpose is the same, however the vectorized code is executed in C, thus is a lot faster.

Pseudo Code

Here is the pseudo code for the question. It is the answer to question number 1, and has been added to the code as well.

1. Define the axes and the meshgrid.
2. Define phi as angle from 0 to 2*π. Using it, find x and y coordinates of elements on the loop
3. Find r' and dl for each element. We find the x and y coordinates and concatenate them. r' is in direction of $\hat{\phi}$, while dl perpendicular to it.
4. Find the current elements, which is a function of Cos(ϕ). Then we plot location of the current elements and the current element (physics) vectors. We plot the (physics) vectors using quiver.
5. Function Calc(paramater l: index): l represents the index of the current element.
    (a) Find x, y and z coordinates of R. We do this, for example with R_x as rx-Loopx[l].
    (b) Now, find magnitude of R as $\sqrt{(R_x^2 + R_y^2 + R_z^2)}$.
    (c) Now, evaluate Ax and Ay as described by the formula in the question. Then return these params.
6. For every element in loop, We call Calc(l) and sum its contribution towards Ax,Ay
7. Now we evaluate B along z axis by using values of Ax and Ay from surrounding cell.

    I have decided to use this as my base for operation. Δx and Δy are equal to 1, so when substituted in the equation - (5) it works. We can simply sum these 4 values and divide by 2 to get the aforementioned equation.

8. Plot the loglog plot of B as a function of z.

9. Use lstsq to try and approximate B(z) to an equivalent of cz^b. We do this by taking log of both sides
   (a) Concatenate log(z) and ones to get m.
   (b) Set n to log(B).
   (c) Then we obtain b, log(c) as output of lstsq(m,n)[0]. Obtain c by taking exponential of log(c).
10. Lastly, plot the fit of B along the z axis with the actual output.
11. Change the value of current, and repeat steps 4 - 10 twice. (Once for each type of current)

Solutions

Question 1

Pseudo code has been included above.

Question 2

The 3D space can be into the prescribed mesh using the meshgrid command. Note that the parameter indexing is set to 'ij'

Listing 1: Question 2

```
x=linspace(-1,1,3)
y=linspace(-1,1,3)
z=linspace(1,1000,1000)
# We create a mesh grid, as asked in the question. We set the indexing to type
ij, as the default goes as y,z,x.
rx,ry,rz=meshgrid(x,y,z,indexing='ij')
```

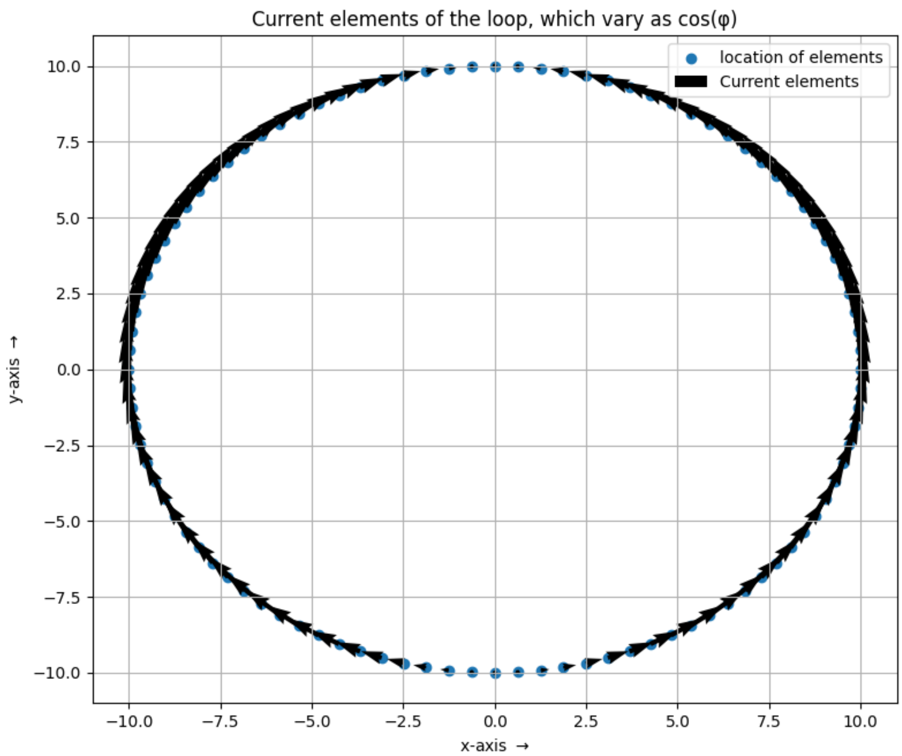
Here, we define phi and using it we define x and y coordinates of the loop. Then, concatenate these x and y coordinates to get r' . Also, we know that dl is perpendicular to r', and is of lenght $2\pi r/100$. So we can evaluate dl as well. Now we define I to be equal to 10**7*Cos(ϕ), and plot the required graphs.

Question 3 and 4

Listing 2: Question 3 and 4

```
phi=linspace(0,2*pi,101)[:-1] # The angle phi as it goes from 0 to 2pi
# over the circumference of the loop.
x_loop=10*cos(phi) # x coordinates of the loop
y_loop=10*sin(phi) # y coordinates of the loop
```

```
# I have done the plots after Question 4.


# Question 4
rdash=c_[x_loop,y_loop]
dl_x=-(pi/5)*(y_loop/10)
dl_y=(pi/5)*(x_loop/10)
dl=c_[dl_x,dl_y]
# We use numpy.c_ to concatenate the x_loop and y_loop vectors for r' and dl_x and dl_y
# for dl vector.
# As length of each dl element should by 2*pi*r/100,
# it has been simplified as such. Note that x_loop and y_loop contain a 10 factor
# within them, and r = 10.

# Variation of current as a function of the angle,
# as described in the question 3 and plot it.
I=(10**7)*cos(phi)

plt.figure(0)
plt.title("Current elements of the loop, which vary as cos(\u03C6)*e^(-jkR)")
plt.scatter(x_loop,y_loop,label="location of elements")
plt.quiver(x_loop,y_loop,dl_x*I,dl_y*I,label = "Current elements")
plt.grid()
plt.legend(loc="upper right")
plt.ylabel("y-axis $\\rightarrow$")
plt.xlabel("x-axis $\\rightarrow$")
plt.show()
```



Question 5 and 6

The function Calc(l) takes the index of the element as a parameter and evaluates Ax and Ay based on said element. First we calculate R_{ijkl}, and use this to find Ax and Ay. Note that here k = 0.1.

Listing 3: Question 5 and 6

```
    l_x,l_y=rdash[l] # x any y coordinates of a particular element in r'

    # Now, let us calculcate |x-l_x| and |y-l_y|, which can be used to
    # finally find R using R = |r-r'|.
    Rx=abs(rx-l_x)
    Ry=abs(ry-l_y)
    Rz=rz
    Rijkl = sqrt(Rx**2+Ry**2+Rz**2)

    # Question 6, compute A vector. k = 0.1
    Axl=cos(2*pi*l/100)*np.exp(complex(0,1)*0.1*Rijkl)*dl[l][0]/Rijkl
    Ayl=cos(2*pi*l/100)*np.exp(complex(0,1)*0.1*Rijkl)*dl[l][1]/Rijkl

    return (Axl,Ayl)
```

Question 7

Evaluate Ax and Ay for the entire 3D mesh, caused by each element in the loop. As A_x and A_y are 3 dimensional matrices, we cannot vectorized them. Instead we resort to using a for loop. As we are only performing this operation 100 times, its fine.

Listing 4: Question 7

```
Ax,Ay=Calc(0)

for l in range(1,100):
    dAx,dAy=Calc(l)
    Ax=Ax+dAx
    Ay=Ay+dAy
```

Question 8

We Find the magnetic Field along the z-axis using the formula below.

$$B_z = \left(\frac{\partial Ay}{\partial x} - \frac{\partial Ax}{\partial y}\right) \implies \frac{Ay(\Delta x, 0, z) - Ay(-\Delta x, 0, z)}{2\Delta x} - \frac{Ax(0, \Delta y, z) - A(0, -\Delta y, z)}{2\Delta y} \quad (6)$$

Listing 5: Question 8

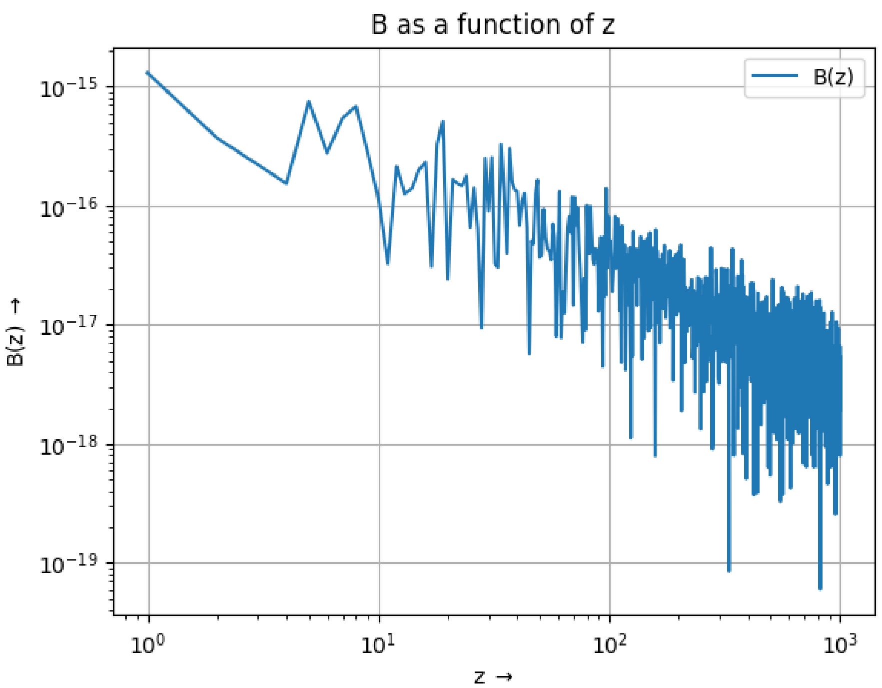
```
B = (Ax[1,0,:]-Ax[1,2,:])/2+(Ay[2,1,:]-Ay[0,1,:])/2
```

Question 9

Here, we plot B as a function of z

Listing 6: Question 9

```
plt.figure(1)
plt.grid()
plt.title("B as a function of z")
plt.loglog(z,abs(B),label="B(z)")
plt.ylabel("B(z) $\\rightarrow$")
plt.xlabel("z $\\rightarrow$")
plt.legend()
plt.show()
```



Question 10

We use the lstsq method to find the closest approximation of B(z) as cz^b. Thus we try to evaluate c and b.

Listing 7: Question 10

```
m = c_[log(z),ones(len(z))] # Set m to [log(z),1]
n = log(abs(B)) # Set n = log(B)
p = lstsq(m,n,rcond=None)[0]
b = p[0]
c = exp(p[1])
bzfit = c*(z**b)
print("For a non static current that varies as a function of cos(phi), the output is")
print("Value of b: ",b)
print("Value of c: ",c)

# Plot B as a function of z along with the approximation from lstsq.
plt.figure(2)
plt.title("B as a function of z, with the best lstsq fit")
plt.grid()
plt.ylabel("B(z) $\\rightarrow$")
plt.xlabel("z $\\rightarrow$")
plt.loglog(z,abs(B),label="B(z)")
plt.loglog(z,bzfit,label="lstsq fit")
plt.legend()
plt.show()
```

For a dynamic current that varies as a function of cos(phi), the output is

Value of b: -0.9202427154359322

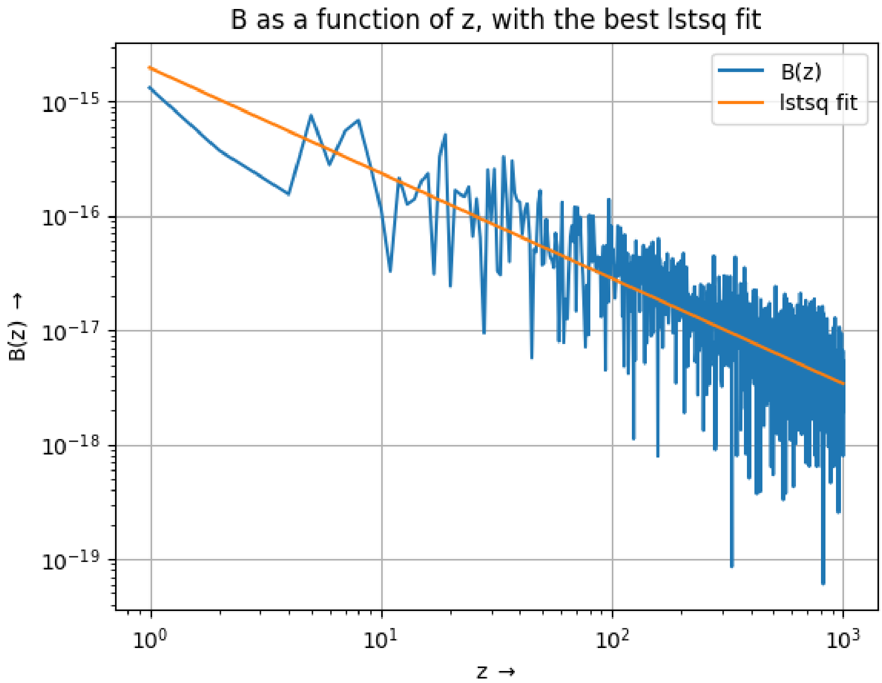
Value of c: 1.959915708500208e-15

Question 11

We get junk values, where is c is nearly negligible. This is expected, as magnetic field along the z axis for such a type of current should be 0. Let us plot this anyway.

Listing 8: Question 11

```
plt.figure(2)
plt.title("B as a function of z, with the best lstsq fit")
plt.grid()
plt.ylabel("B(z) $\\rightarrow$")
plt.xlabel("z $\\rightarrow$")
plt.loglog(z,abs(B),label="B(z)")
plt.loglog(z,bzfit,label="lstsq fit")
plt.legend()
plt.show()
```

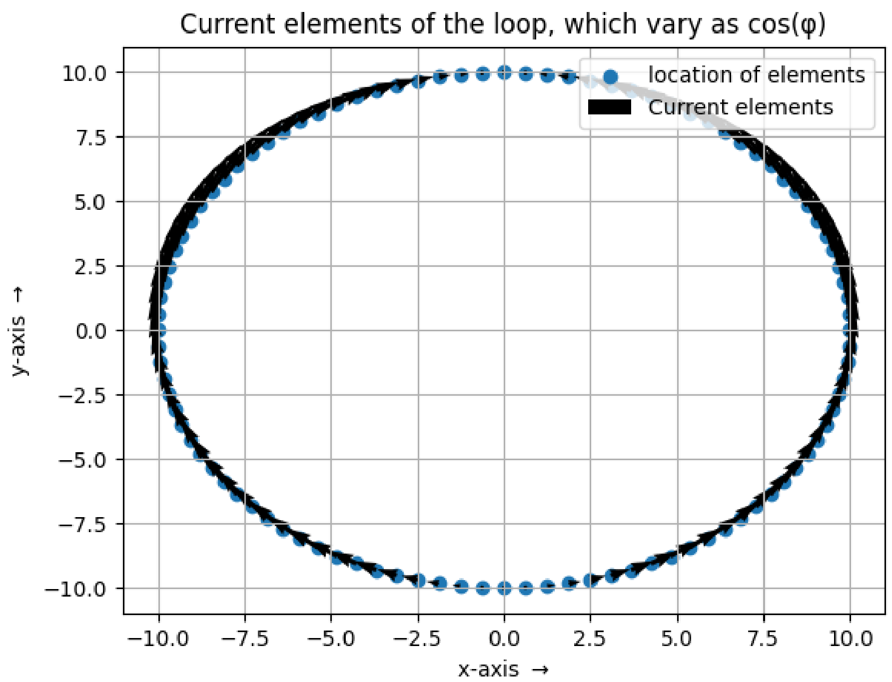


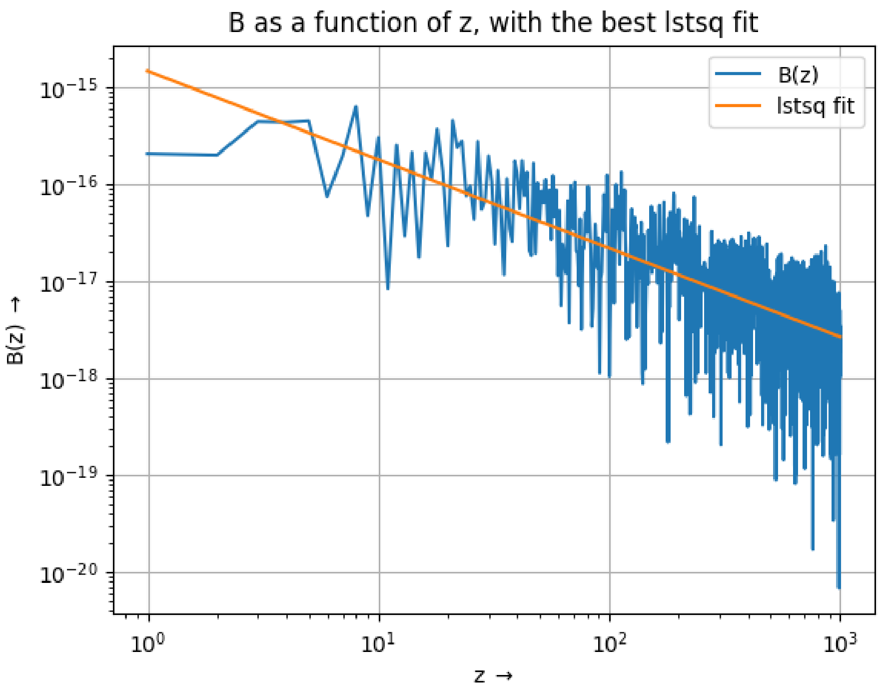
This is not very useful. Let us try again with a static current. Only the current term and the definition of the Calc function changes. We set k=0.

Listing 9: Question 11: Continued

```
I=(10**7)*cos(phi)

def Calc2(l):
    l_x,l_y=rdash[l] # x any y coordinates of a particular element in r'

    # Now, let us calculcate |x-l_x| and |y-l_y|, which can be used to finally
    # find R using R = |r-r'|.
    Rx=abs(rx-l_x)
    Ry=abs(ry-l_y)
    Rz=rz
    Rijkl = sqrt(Rx**2+Ry**2+Rz**2)

    Axl=cos(2*pi*l/100)*dl[l][0]/Rijkl
    Ayl=cos(2*pi*l/100)*dl[l][1]/Rijkl

    return (Axl,Ayl)
```

On the next page are the relevant plots that we get for such a case. Still, c is negligible, and still the reason is the function that we used to define current.





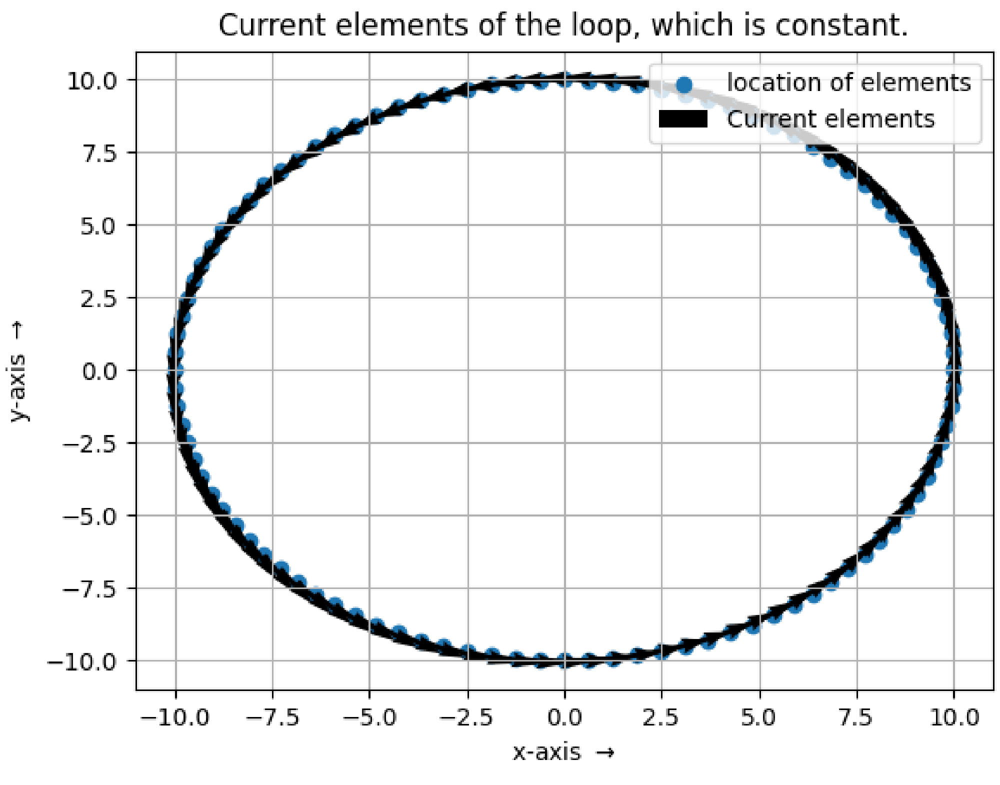
For a static current that varies as a function of cos(phi), the output is Value of b: -0.9143206583596813

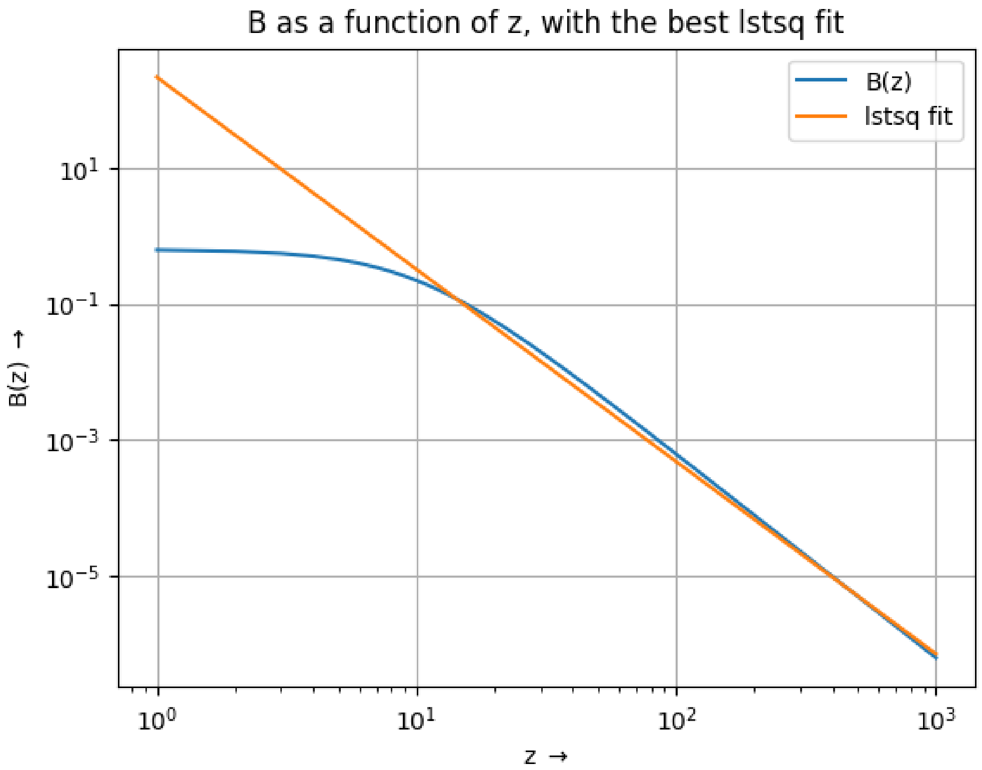
Value of c: 1.462527559494362e-15

Let us try one last time with with a constant current.

Listing 10: Question 11: Continued: Again

```
I=(10**7)

def Calc3(l):
    l_x,l_y=rdash[l] # x any y coordinates of a particular element in r'

    # Now, let us calculcate |x-l_x| and |y-l_y|, which can be used to finally
    # find R using R = |r-r'|.
    Rx=abs(rx-l_x)
    Ry=abs(ry-l_y)
    Rz=rz
    Rijkl = sqrt(Rx**2+Ry**2+Rz**2)

    Axl=dl[l][0]/Rijkl
    Ayl=dl[l][1]/Rijkl

    return (Axl,Ayl)
```





For a static current that is constant, the output is

Value of b: -2.826192056926662

Value of c: 215.85790244343454

Success! We have got a non zero value of B along the z axis for using such a current. The reason we got some value in the previous 2 cases was due to a precision error, and finally we have got some value. For larger values of z, B_z seems to mirror something close to c*z^{-3}. The decay rate is -2.82.

Conclusion:

We have successfully calculated the magnetic field along the z-axis for various types of current. We found this magnetic field by evaluating the curl of the magnetic potential. Then, we used the lstsq method, to find the decay rate. For a dynamic case, we got a decay rate of -0.92 and for the static case, a decay rate of -0.91. However these 2 values are not very useful, as we should ideally get 0 as the magnetic field along the z-axis in both these cases. So upon inspection of a third type of current that is constant, we get a decay rate of -2.82. Thus we have concluded this Assignment.