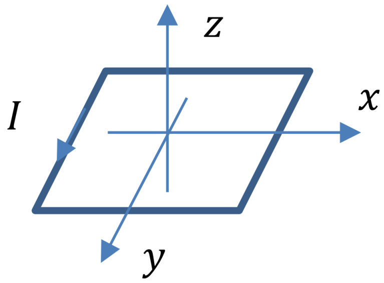
HW C Writeup

PHYS3602, Northeastern University
Justin Vega
December 6th, 2019

1 Problem:

HW #7, Q2 (with modification)
Consider a square loop of current in the xy-plane with sides of length L with its center at the origin.



Original: Find the magnetic field on the center axis (xy-axis) as a function of z.

Modification: Visualize the magnetic field on and off the center axis for y = 0, as a function of x and z (x and y axis are switched from original homework problem)

Tools Used: Python

2 Mathematical Methods

For points (0, z), the following equation in the homework was used, with $\mathscr{r} = \sqrt{z^2 + \frac{L^2}{2}}$, $\theta_1 = \frac{-L}{2\mathscr{r}}$ and $\theta_1 = \frac{L}{2\mathscr{r}}$.

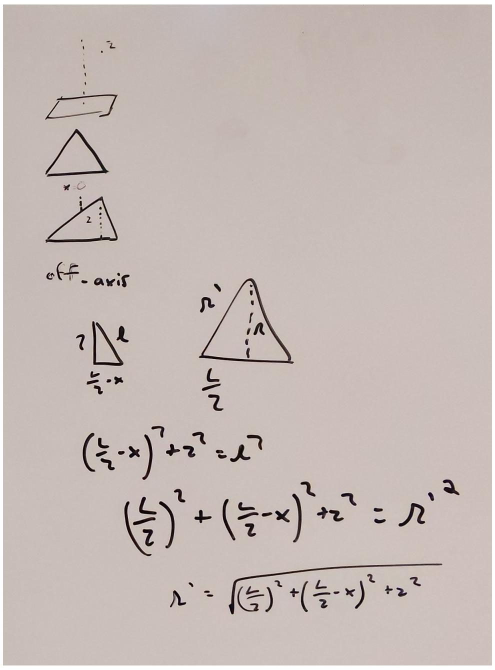
$$B = \frac{\mu_0 I}{4\pi \mathscr{r}}(\sin\theta_2 - \sin\theta_1)$$

To obtain the z component of the field, a factor of cos(θ), with $\theta = \frac{L}{2\sqrt{z^2+L^2/4}}$ was multiplied to the B field to obtain B_z.

For points (x, z), the field was calculated separately as a function of x and z for the left and right wires running parallel to the y-axis. Figures 1-3 show the relevant math used to obtain the $\mathscr{r}$, θ_1, and θ_2.

The left and right magnetic fields were broken down into B_x and B_z by equating $B_x = B\sin\theta$ and $B_z = B\cos\theta$, with $\theta = \arccos(\frac{L/2 \pm x}{\mathscr{r}})$, [left (+) right (-)]. Finally, both left and right B_x and B_z were added to obtain the magnitude of the magnetic field at a point.

For points (x, 0), the field was calculated the same as (x, z); the equations are still valid.

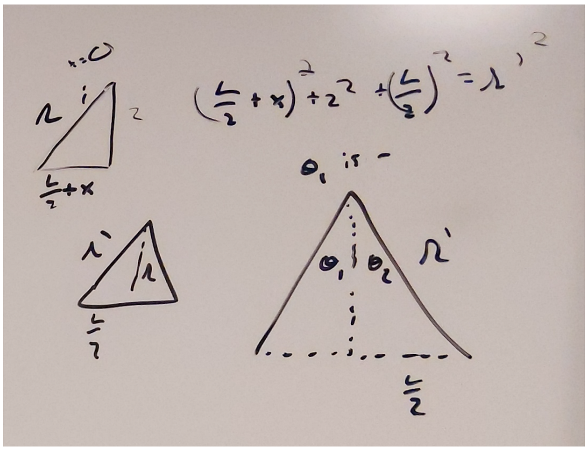


$$\left(\frac{L}{2}-x\right)^2+z^2=l^2$$

$$\left(\frac{L}{2}\right)^2+\left(\frac{L}{2}-x\right)^2+z^2=r'^2$$

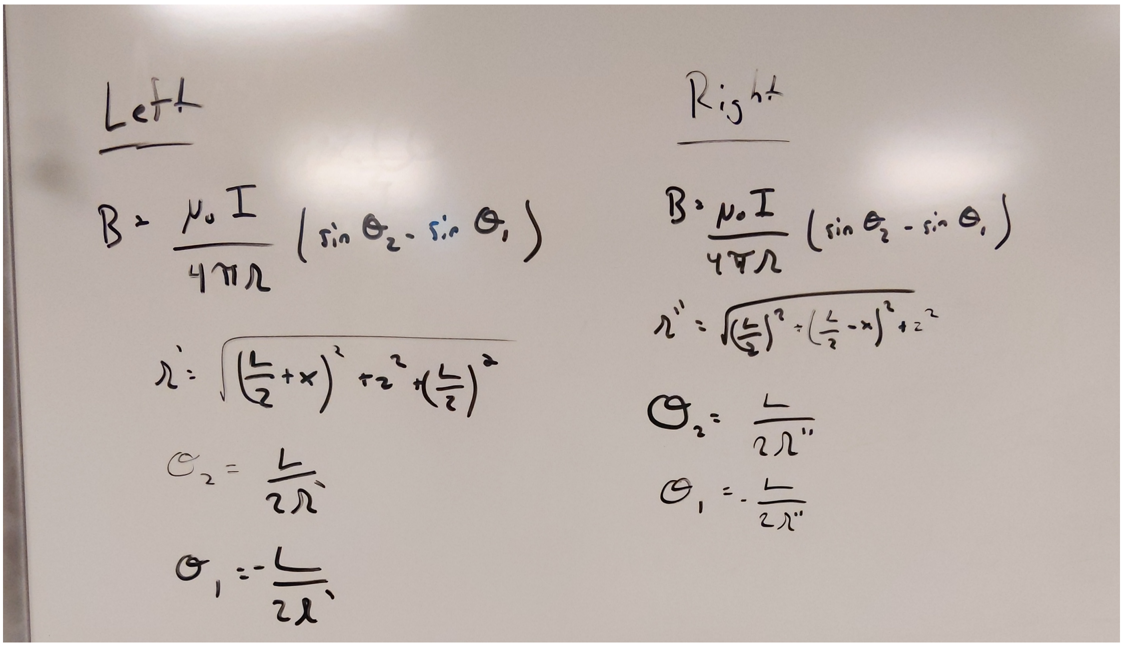
$$r'=\sqrt{\left(\frac{L}{2}\right)^2+\left(\frac{L}{2}-x\right)^2+z^2}$$

Figure 1: Left



$$\left(\frac{L}{2}+x\right)^2+z^2+\left(\frac{L}{2}\right)^2=r'^2$$

Figure 2: Right



Left

$$B=\frac{\mu_0 I}{4\pi r}\left(\sin\theta_2-\sin\theta_1\right)$$

$$r'=\sqrt{\left(\frac{L}{2}+x\right)^2+z^2+\left(\frac{L}{2}\right)^2}$$

$$\theta_2=\frac{L}{2r'}$$

$$\theta_1=-\frac{L}{2l'}$$

Right

$$B=\frac{\mu_0 I}{4\pi r}\left(\sin\theta_2-\sin\theta_1\right)$$

$$r''=\sqrt{\left(\frac{L}{2}\right)^2+\left(\frac{L}{2}-x\right)^2+z^2}$$

$$\theta_2=\frac{L}{2r''}$$

$$\theta_1=-\frac{L}{2r''}$$

Figure 3: All Equations

3 Code

The Python script added at the end of this writeup generates a coordinate grid with $-L \leq x \leq L, -10 \leq z \leq 10$ and calculates the magnetic field for every single coordinate. The user is asked to specify length and current values for the calculation.

3.1 Example Output

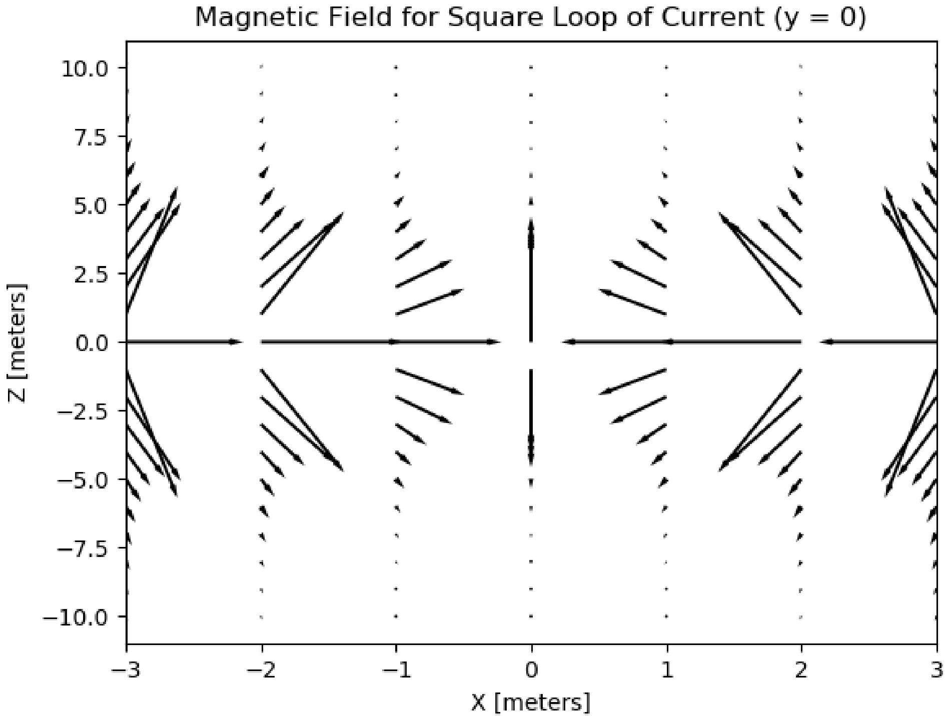


Figure 4: Output with $I = 3A$, $L = 3$ meters

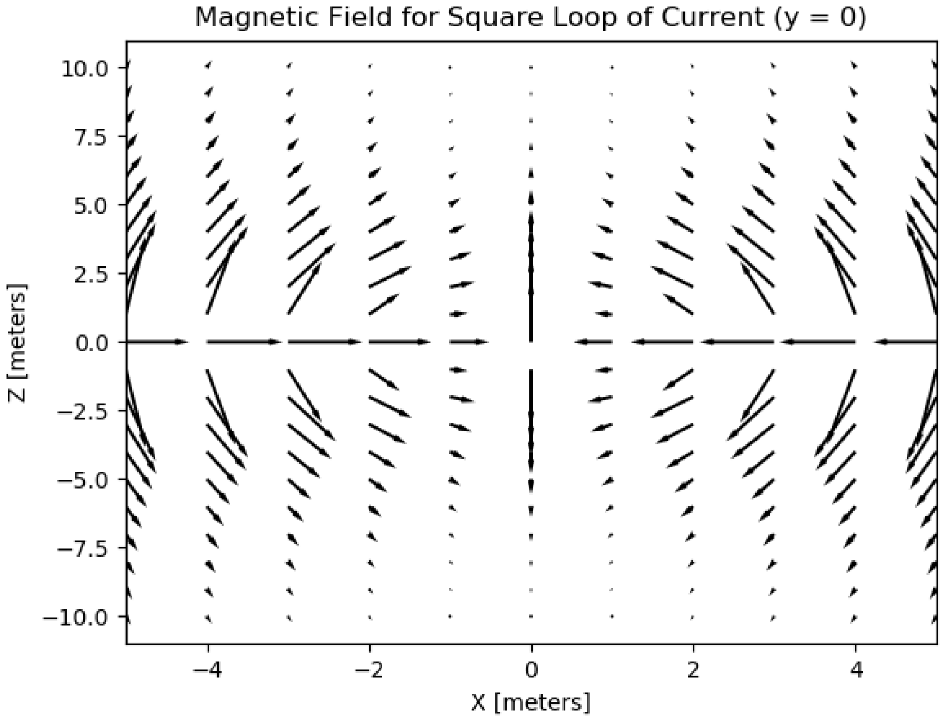


Figure 5: Output with $I = 2A$, $L = 5$ meters

3.2 Python Code

```
import math
import numpy as np
import matplotlib.pyplot as plt
import scipy.constants as sp
import re

usercurrent = (input("What␣would␣you␣like␣the␣current␣to␣be,␣in␣amps
    ?␣"))
userlength = (input("What␣would␣you␣like␣the␣length␣of␣one␣side␣of␣
    the␣wire␣to␣be,␣in␣meters?␣"))

I = int(re.sub("[^0-9]", "", usercurrent))
L = int(re.sub("[^0-9]", "", userlength))

print("You␣want␣%s␣amps␣of␣current␣and␣the␣length␣of␣one␣side␣of␣the
    ␣square␣loop␣to␣be␣%s␣meters." % (I, L))
print("Great!␣Calculating␣...")

mag = [] #list of x, z magnitudes

def biosavart(x1, z1): #mag field calculation
    l_scr = np.sqrt((L/2+x1)**2+z1**2+(L/2)**2) #script r, x<0
    l_theta1 = -L/l_scr #theta1, x<0
    l_theta2 = L/l_scr #theta2, x<0
    l_biosavart = ((mu*I)/(4*math.pi*l_scr)*(math.sin(l_theta2)-math
        .sin(l_theta1))) #mag field from x<0, in Tesla
    lx_biosavart = l_biosavart * (z1/l_scr) #mag field from x<0, x
        contribution
    lz_biosavart = l_biosavart * ((L/2+x1)/l_scr) #mag field from x
        <0, z contribution
    r_scr = np.sqrt((L/2-x1)**2+z1**2+(L/2)**2) #script r, x>0
    r_theta1 = -L/r_scr #theta1, x>0
    r_theta2 = L/r_scr #theta2, x>0
    r_biosavart = -((mu*I)/(4*math.pi*r_scr)*(math.sin(r_theta2)-
        math.sin(r_theta1))) #mag field from x>0, in Tesla
    rx_biosavart = r_biosavart * (z1/r_scr) #mag field from x>0, x
        contribution
    rz_biosavart = r_biosavart * ((L/2-x1)/r_scr) #mag field from x
        >0, z contribution
    if z1 < 0:
        lz_biosavart = -(lz_biosavart) #makes z component negative
        rz_biosavart = -(rz_biosavart) #makes z component negative
    if x1 < 0:
        lx_biosavart = -(lx_biosavart) #makes x component negative
        rx_biosavart = -(rx_biosavart) #makes x component negative
        lz_biosavart = -(lz_biosavart) #makes z component negative
        rz_biosavart = -(rz_biosavart) #makes z component negative
    if x1 == 0: #filtering out solely (0, z) magnetic fields ,
      simpler
        mag.append((0, lz_biosavart)) #B_z = one side's z
          contribution
        return
    elif z1 == 0: #filtering out solely (x, 0) magnetic fields ,
      simpler
        mag.append((l_biosavart+r_biosavart , 0)) #B_x = sum of x
          components from both directions
        return
    elif x1 < 0: #−x calculations
        mag.append((abs(lx_biosavart+rx_biosavart), (lz_biosavart+
          rz_biosavart)))
        return
    elif x1 > 0: #x calculations
        mag.append(((−abs(lx_biosavart+rx_biosavart)), lz_biosavart+
          rz_biosavart))
        return

#generates x pairs of random integer coordinates from x to x
coords = [(x,y) for x in range(−L, L+1) for y in range(−10, 11)]
mu = sp.mu_0

x1points = [] #solely generated x coordinates
z1points = [] #solely generated z coordinates

for idx, pair in enumerate(coords):
    x1, z1 = pair
    x1points.append(x1)
    z1points.append(z1)
    biosavart(x1, z1) #run the function

plotx = [] #total x magnitude of field
plotz = [] #total z magnitude of field

for idx, points in enumerate(mag): #tuple isolation by (x, z)
    magx, magz = points
    plotx.append(magx)
    plotz.append(magz)

#plots x, z generated coordinates with x and z magnitudes
plt.quiver(x1points, z1points, plotx, plotz, width = 0.004,
  headwidth = 2, headlength = 4)
plt.xlim(−L, L)
plt.title('Magnetic Field for Square Loop of Current (y = 0)')
plt.xlabel('X [meters]')
plt.ylabel('Z [meters]')
plt.show()
```