BiotSavart1-Q-Py

June 22, 2021

[1]:
```
#if you want this notebook to use the noTeXbook theme
#install with $pip install notexbook-theme in a terminal
#and then uncomment the next two lines:
#%load_ext notexbook
#%texify
```

[1]:
```
<IPython.core.display.HTML object>
```

1 Biot Savart Law I

1.1 On-axis field of a current loop

In this exercise, you will be computing the magnetic field for a wire loop of radius R carrying current I, both on the axis of symmetry and off the axis of symmetry. The former can be computed analytically with relative ease, and although the latter can also, the solution involves the complete elliptic functions of the first and second kind (you'll solve this in graduate school should you choose to continue your study of physics).

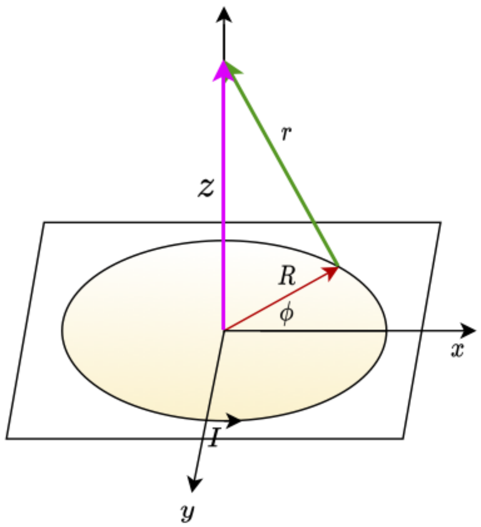
Estimated time to complete: 2-4 hours

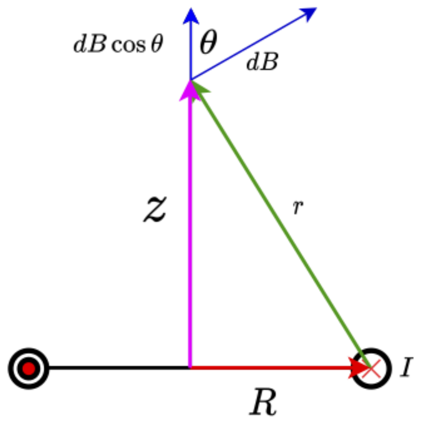
1.2 Theory: Derivation of the on-axis field

For the first part of this exercise, we want you to use the Biot-Savart Law

$$d\vec{B} = \frac{\mu_o I}{4\pi}\frac{\vec{dl} \times \hat{r}}{r^2}$$

to compute the magnetic field on the axis of symmetry of a loop with radius R a distance z above the center of the loop; the relevant geometry is shown in the figure below.





Side view of 3d figure at $\phi = 0$

Using a markdown cell explain using LaTeX how to use the Biot-Savart Law to compute the magnetic field on the z-axis. Explain your thinking clearly, and include as the last item, the expression for $B(z)$.

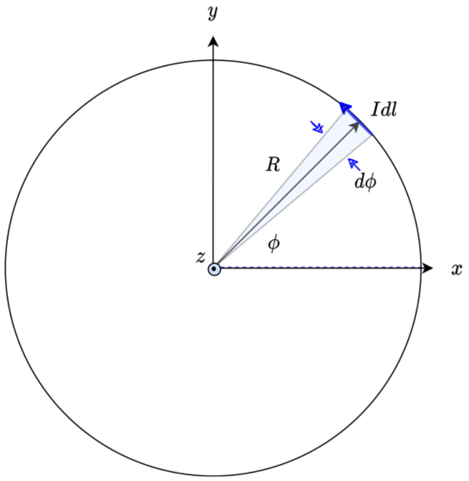
1.3 Numerical: Computation of the on-axis magnetic field

Your task, using Python, is to write a function which computes the on-axis magnetic field in nanoTesla, and make sure to address each of the points (a through e) below:

```
a. use numPy's cross() function to compute the cross product which appears in the
 Biot-Savart Law,
b. include a triple quoted docstring that describes the function,
c. include a test of the function for $z=0$, and make sure it returns the expected
 value (hint: what is the theoretical value?)
d. produce a scatter plot of the field for a loop of radius 1.0 meter carrying a
 current of 1.0 Ampere. Plot the magnetic field in nT for z values from -5 to +5 meters.
e. plot the theoretical curve for B(z) over the same range.
```

1.3.1 Steps to lead you through the computation

The source: Idl In order to compute the magnetic field for a small element of current $I\vec{dl}$, we need to write down the (x, y, z) components of this vector in terms of the loop radius, R, the angular position of the segment ϕ, and the differential length of the segment. So first, looking at the diagram below, explain in a markdown cell (using LaTeX to write the mathematics) how to write $I\vec{dl}$ as a vector with x, y, and z components; your answer should be in terms of R, ϕ, $d\phi$, and I.



In your explanation, you should draw a figure to help explain your thinking. I suggest the app draw.io

The source to field point vector: $\vec{r}$ Now, use a markdown cell to describe the source to field point vector r. This vector points *from* the source of the magnetic field ($I\vec{dl}$) *to* the field point on the z axis (as shown in the first figure). This vector will have components in all three coordinate directions. Write down this vector $\vec{r}$ in the markdown cell below, and convince the reader that you have the correct expression by picking several special cases for the source and field point where you know *a priori* what $\vec{r}$ should be.

Put your explanation here

Putting it into code! Now that you have the 3D vectors $I\vec{dl}$ and $\vec{r}$, write a function called *b_field* which takes as arguments I, R, z, and $\delta\phi$, and returns the three components of the magnetic field at a point z above the center of the loop. We assume that the current is CCW when viewed from $+z$ (as the figures all show). Be careful about the limits of your integration when you compute $\vec{B}$. After writing the code in the cell below, answer (a) through (c) using code and markdown cells. We'll get to (d) and (e) after you complete (a) through (c). NumPy has a function called *cross(a,b)* which computes the cross product of two vectors a and b.

[ ]:

```
# put your code here
import numpy as np
import matplotlib.pyplot as plt

def b_field(I, R, z, ):


    return B
```

Use a series of markdown and code cells to test your b_field function:

1.3.2 Finishing up: Now compute the magnetic field on axis and compare to theory.

Now that you previously derived the theoretical expression for $B(z)$ and the computational solution,

Define a function called b_theory which computes the theoretical value of B for a given input I, R, and z. Test the function to make sure it gives reasonable values. Compute and plot the z-component of the magnetic field from $z = -5$ m to $+5$ m, both theoretically and via simulation (plot as a scatter plot of points). Remember to label your axes!

[ ]:

```
def b_theory(I,R,z):

    return B
```

[ ]:

```
# YOUR TESTS HERE --- convince me that this function gives the correct answer!
```

Compute the magnetic field via numerical simulation and the theoretical method, and compare their values:

[ ]:

```

```

Now plot theory versus numerical computation:

[ ]:

```
# use these plot parameters
plt.rcParams["figure.figsize"] = (12,7)

plt.rcParams.update({
    "text.usetex": True,
    "font.family": "serif",
    "font.serif": ["Times"],
})
plt.xticks(fontsize=16)
plt.yticks(fontsize=16)
```