Python/MATLAB for Physics
USPAS February 2022
Homework 1 Assignment

February 7, 2022

Due: 10am cst, Wed Feb. 9th

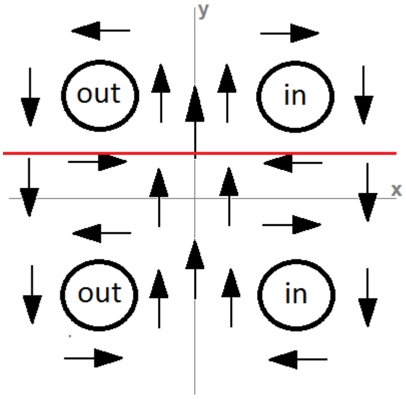
Problem 1.1: Simple Dipole Magnet

The magnetic field from a wire at (x_0, y_0) is given by:

$$\vec{B_x} = \frac{\mu_0 I}{2\pi} \frac{(y - y_0)}{(x - x_0)^2 + (y - y_0)^2}$$
$$\vec{B_y} = -\frac{\mu_0 I}{2\pi} \frac{(x - x_0)}{(x - x_0)^2 + (y - y_0)^2}$$

Where $I > 0$ represents current flow into the $x - z$ plane (i.e. into the page) and $I < 0$ represents current out of the plane.

Now we consider four wires - one at (-1,1), one at (1,1), one as (1,-1), and one as (-1,-1). Let the current be into the plane for the wires on the right side (where x=1) and out of the plane for the wires on the left side (where x=-1), as depicted in the diagram below. The wires generate a vertically aligned magnetic field, forming a simple "horizontal dipole" corrector.



Program

Write a Python program to calculate the magnetic field from in the $x - y$ plane from four wires in the dipole configuration. Let $\frac{\mu_0 |I|}{2\pi} = 1$ and let x and y each go from -2 to 2. Use numpy clip (or max and min) to restrict Bx and By to values between -5 and 5. Make a quiver plot of the magnetic field in $x - y$ plane. In another figure, plot the value of B_x and B_y on a horizontal line just above the center plane with $y = 0.5$ and x from -2 to 2 (shown in read)

Problem 1.2: 4D Gaussian Distribution

The transverse position of a particle can be expressed in four normalized phase-space coordinates X, Px, Y, Px. In a 4D Gaussian distribution, the projection of the distribution on to each plane will itself be a 1D Gaussian distribution. Consequently, if each of the four phase-space coordinates is drawn from an independent 1D Gaussian distribution, then the transverse position is drawn fron a 4D Gaussian distribution.

The transverse position is said to exceed “three sigma“ horizontally if $\sqrt{X^2 + Px^2} > 3$ or vertically if $\sqrt{Y^2 + Py^2} > 3$. A 4D Gaussian distribution that exclude values which exceed three sigma is referred to as a truncated 4D Gaussian distribution.

Program

Write a Python program to generate 10,000 particles each with a transverse phase-space coordinates drawn from a truncated 4D Gaussian distribution. Use a mean of 0 and a sigma of 1 for each 1D Gaussian projections. If the transverse position for any particles exceeds three sigmas either horizontally or vertically, replace that transverse position with a new transverse position drawn from a 4D Gaussian distribution (repeating if necessary until it no longer exceeds three sigmas). For the X-coordinates of the particles, output the mean, the maximum value, the minimum value, and the standard deviation.

Make a 2D histogram plot showing the density of particles in the X-Y plane (disregarding the Px and Py values). The 2D histogram show span from -3 to 3 in X and Y and use 0.1 bins. Make sure to include the colorbar on your plot.

Problem 1.3: FFT Spectrum of FM Signal

Frequency Modulated (FM) waves show up in communication, noise, and diagnostics. In FM communication, a carrier frequency is used to transmit data and a modulation frequency is used to encode data. In accelerator noise, an unwanted frequency modulation in a focusing element (such as quadrupoles or an RF cavity) may induce a modulation in the dynamical frequencies of the beam. In diagnostics, the presence of a time-dependent subject of interest (such as particles in an electron cloud) have a modulating effect on an microwave transmission.

The frequency spectrum of a FM wave can be shown to be given by the carrier frequency plus/minus harmonics of the modulation frequency, with magnitudes given by evaluating Bessel functions:

$$\cos[\omega_c t + B\sin(\omega_m t)] = \sum_{n=-\infty}^{\infty} J_n(B)\cos(\omega_c t + n\omega_m t)$$

Prep

Move the file `HW_data` from the `homework` folder of the `USPAS_Python_2022` public Github repository to your local submodule (to the same directory where you will run your Python code).

The file `HW_data` is a space-separated CSV file that contains a 1D float array representing data to be analyzed. The following code snippet can be used to put the data to be analyzed into a 1D numpy array `x`:

```
import numpy as np
import csv

with open('HW_data') as csv_file:
  csv_reader = csv.reader(csv_file, delimiter=' ')
  for row in csv_reader:
    if len(row) > 0:
      x = np.array(row).astype(float)
```

This code snippet can be found in the file `HW_data_input.py`. The code used to generate `HW_data` in the first place is `HW_data_generator.py` located in the same directory of the public Github repository.

Program

Write a program to calculate the FFT spectrum of the `x` from the file `HW_data` (see Prep section above). Make a plot of the (absolute magnitude) of the FFT spectrum with tune on the horizontal axis and amplitude on the vertical axis.

Assume that x represents a timeseries with each element separated by 1 μs. The program should calculate and output an estimate of the carrier frequency and modulation frequency derived from the FFT spectrum (some amplitude and phase noise has been added to the data).