TotalMagneticField.py

```
import numpy as np
import matplotlib.pyplot as mp
from scipy.integrate import quad


############################################################
# Function to get the Magnetic Field for a particular x
############################################################
def getYMagneticFieldAtX(x, alpha):
    def integral(theta):
        return np.sin(theta)/(((x**2) - 2*x*np.cos(alpha)*np.sin(theta))**1.5)
    res, err = quad(integral, 0, 2*np.pi)
    if (x * np.sin(alpha) * res) < 0.000001:
        return 0
    return x * np.sin(alpha) * res

def getZMagneticFieldAtX(x, alpha):
    def integral(theta):
        return (1 - x*np.cos(alpha)*np.sin(theta))/(((x**2) - 2*x*np.cos(alpha)*np.sin(theta))**1.5)
    res, err = quad(integral, 0, 2*np.pi)
    return res

# x is the ratio of rho and a
def plotPartB(alpha, start = 1, end = 5, stepsize = 0.1):
    xAxis = [start + i*stepsize for i in range(int(end//stepsize) + 2)]
    yAxis = [getYMagneticFieldAtX(xValue, alpha) for xValue in xAxis]
    zAxis = [getZMagneticFieldAtX(xValue, alpha) for xValue in xAxis]
    mp.plot(xAxis, yAxis, label = "Y component")
    mp.plot(xAxis, zAxis, label="Z component", ls = ":")
    mp.xlim(xmin=1.5)
    mp.legend()
    mp.xlabel('X/a ( > 1.5)')
    mp.ylabel('B/Bnot')
    mp.title('B v/s X. Alpha = PI/2')
    mp.show()
    #mp.savefig('By v/s X')

#print (getMagneticFieldAtX(1.5, np.pi/2))

#plotPartB(0, 1.5, 10, 0.01)
#plotPartB(np.pi/4, 1.5, 10, 0.01)
plotPartB(np.pi/2, 1.5, 10, 0.01)
```