PHYSICS PROJECT

PHYSICS EXPERIMENT CALCULATOR

THAPAR INSTITUTE
OF ENGINEERING & TECHNOLOGY
(Deemed to be University)

Submitted by:
Vanshaj Raghuvanshi – 102216111 – CS12
Jatin Aggarwal – 102216084 – CS11
Naman Babbar – 102216116 – CS12
Saiyam Singhi - 102216108 – CS12
Rohan Purohit - 102216120 – CS12
Sayak Mukherjee - 102216095 – CS12

Submitted to:
Dr. Rajkumar Khan

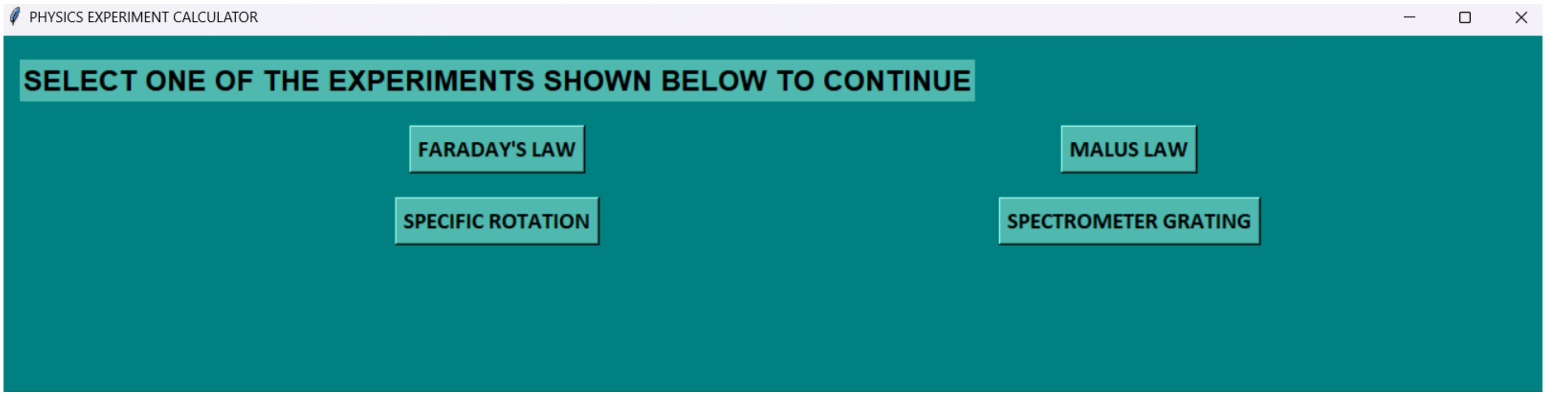
ABSTRACT

Physics experiments in the first year of college play an imperative role in students’ academics, and their understanding of the concepts they are taught. They conduct these experiments and note down the readings they obtain, but they have to give precious time towards writing down these experiments in their practical files, and plotting the graphs of specific experiments and doing calculations. This project is a program made in python, which will do the calculation and plotting of graph for the students, which will in turn save time and effort.

OBJECTIVE

To provide a program which does the calculations and plotting of graphs of physics experiments, taking the readings as input.

MAIN SCREEN



GUI based interface to select the experiment of choice

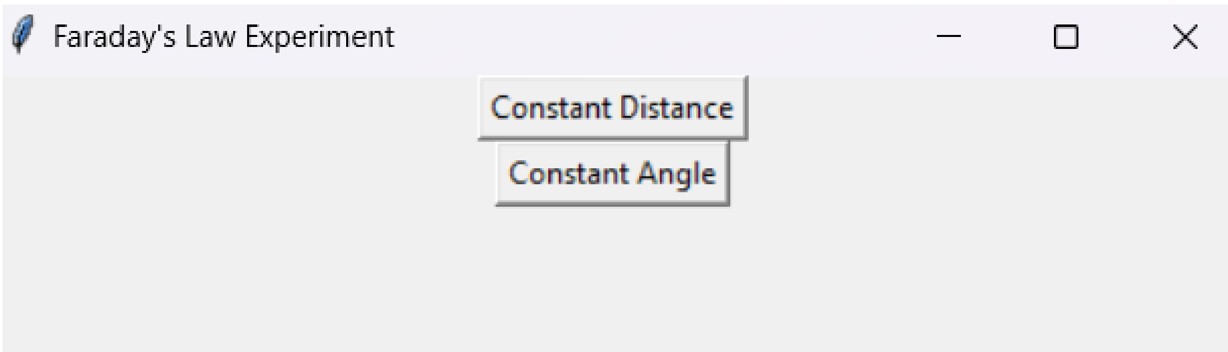
Experiment-1 : FARADAY’S LAW

In this experiment, the goal is to investigate how the induced electromotive force (emf) in a conductor changes as a function of the velocity of a magnet moving near the conductor. To perform the experiment, a magnet is mounted on a cart that can move back and forth along a track. The magnet is positioned near a coil of wire, which forms the secondary circuit of the experiment.

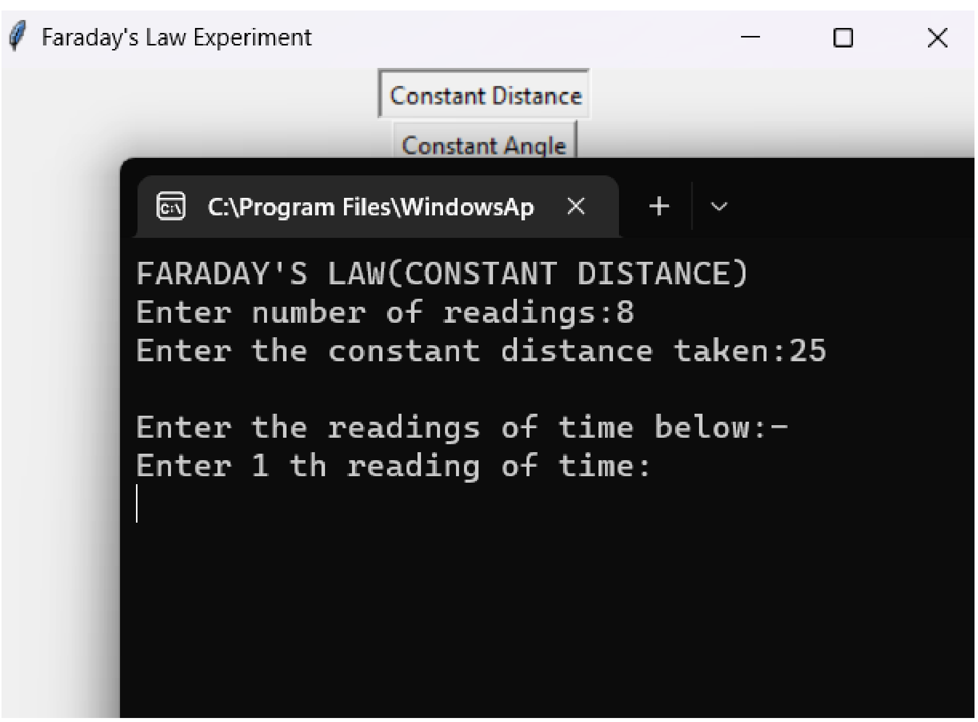
As the magnet is moved back and forth along the track, the magnetic field around the coil changes, inducing an emf in the secondary circuit. The magnitude and direction of the induced emf can be measured using a galvanometer or an ammeter, which are connected in parallel or series with the secondary circuit, respectively.

To determine how the induced emf changes as a function of the magnet's velocity, the cart is moved back and forth along the track at different speeds. The induced emf is measured at each velocity and the results are recorded. This allows the experimenter to plot a graph of induced emf versus velocity and observe the relationship between these two variables.
It is expected that the induced emf will increase as the velocity of the magnet increases. This is because a higher velocity will result in a larger change in the magnetic field around the coil, which will in turn induce a larger emf in the secondary circuit. By performing the experiment at a range of velocities and analyzing the results, the experimenter can gain a better understanding of the relationship between velocity and induced emf. This type of experiment is useful for demonstrating the principles of electromagnetic induction and for exploring the behavior of electric circuits.
In this experiment we have 2 cases:
1.TAKING ANGLE CONSTANT
2.TAKING DISTANCE CONSTANT
In both cases we have calculated time period after 10 oscillations of the magnet.
noted the voltage and calculated the maximum velocity using formula
$V_{max}=(4\pi/T)*\sin(\Theta/2)$
At last we have plotted the graph between emf and velocity(max).
We have used math module to calculate sinx and for value of pi, matplotlib module for plotting graph,use of numpy module for coordinates, tkinter module and its functions for gui output
we take distance , angle, time and induced emf as input and provide the maximum velocity and the graph between induced emf and maximum velocity as output.

OUTPUT AND INPUT SCREENS FOR OPTION OF FARADAY'S LAW



CONSTANT DISTANCE:-



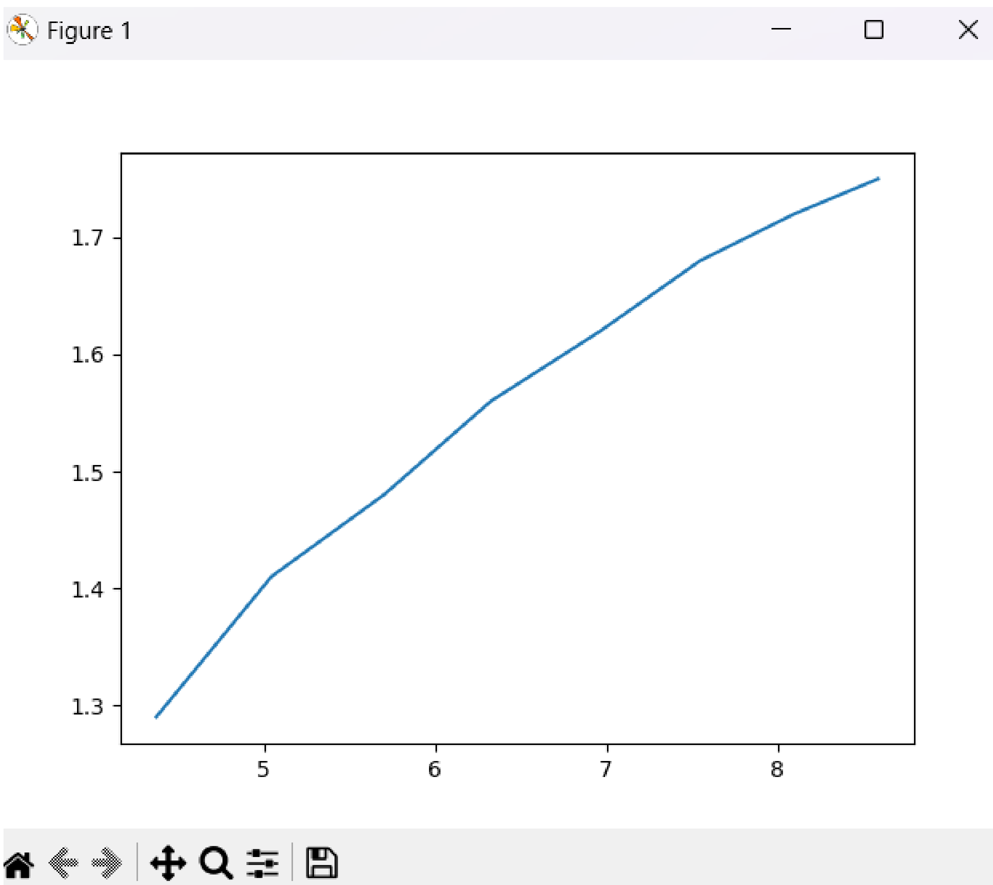
```
Enter the constant distance taken:25

Enter the readings of time below:-
Enter 1 th reading of time:
19.66
Enter 2 th reading of time:
19.4
Enter 3 th reading of time:
19.22
Enter 4 th reading of time:
19.06
Enter 5 th reading of time:
19
Enter 6 th reading of time:
18.84
Enter 7 th reading of time:
18.72
Enter 8 th reading of time:
18.59

Enter the readings of angle below:-
Enter 1 th reading of angle:
65
Enter 2 th reading of angle:
60
Enter 3 th reading of angle:
55
Enter 4 th reading of angle:
50
Enter 5 th reading of angle:
45
Enter 6 th reading of angle:
40
Enter 7 th reading of angle:
35
Enter 8 th reading of angle:
30

Enter the readings of peak emf below:-
Enter 1 th reading of peak emf:
```

```
Enter the readings of peak emf below:-
Enter 1 th reading of peak emf:
1.75
Enter 2 th reading of peak emf:
1.72
Enter 3 th reading of peak emf:
1.68
Enter 4 th reading of peak emf:
1.62
Enter 5 th reading of peak emf:
1.56
Enter 6 th reading of peak emf:
1.48
Enter 7 th reading of peak emf:
1.41
Enter 8 th reading of peak emf:
1.29
```

RESULTS

Velocities in order:-

8.585841822782577
8.09688828244792
7.547482055902482
6.965867924225969
6.327555051414171
5.70322701513614
5.046448347299518
4.373878486838109



CONSTANT ANGLE:-

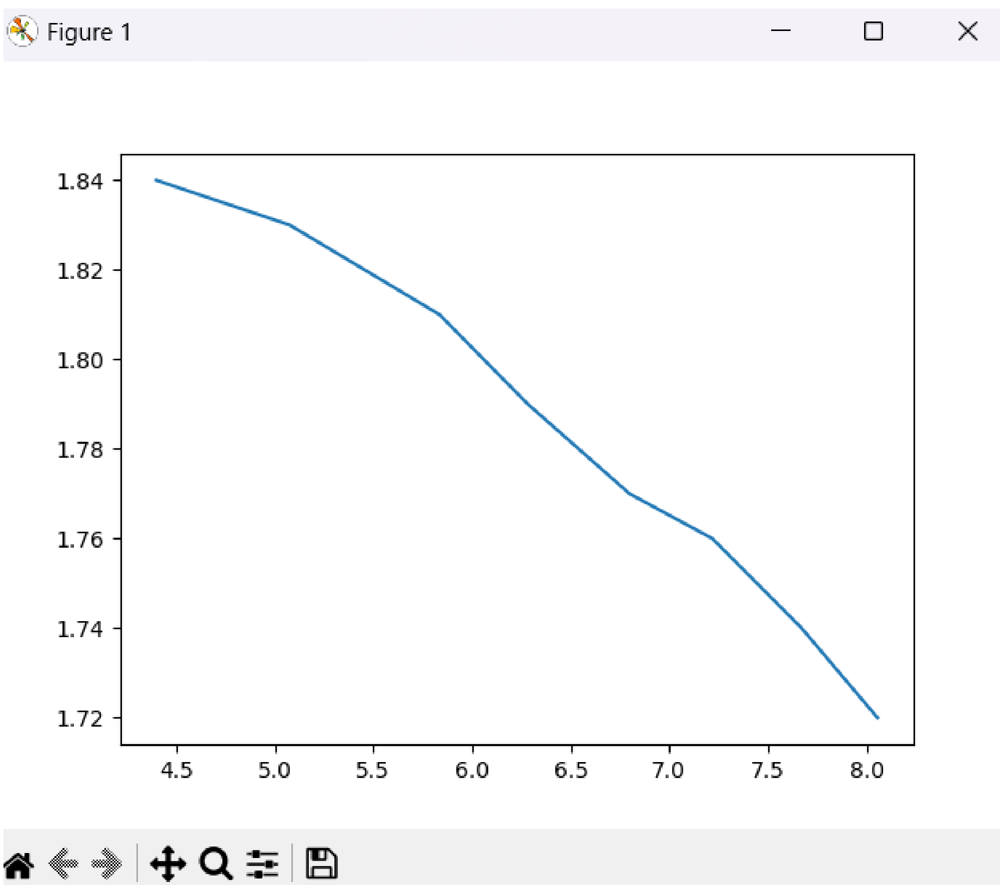
```
FARADAY'S LAW(CONSTANT ANGLE)
Enter number of readings:8
Enter the constant angle taken:60

Enter the readings of time below:-
Enter 1 th reading of time:
19.5
Enter 2 th reading of time:
18.84
Enter 3 th reading of time:
18.28
Enter 4 th reading of time:
17.56
Enter 5 th reading of time:
17
Enter 6 th reading of time:
16.15
Enter 7 th reading of time:
16.09
Enter 8 th reading of time:
15.71

Enter the readings of distance below:-
Enter 1 th reading of distance:
25
Enter 2 th reading of distance:
23
Enter 3 th reading of distance:
21
Enter 4 th reading of distance:
19
Enter 5 th reading of distance:
17
Enter 6 th reading of distance:
15
Enter 7 th reading of distance:
13
Enter 8 th reading of distance:
11
```

```
Enter the readings of peak emf below:-
Enter 1 th reading of peak emf:
1.72
Enter 2 th reading of peak emf:
1.74
Enter 3 th reading of peak emf:
1.76
Enter 4 th reading of peak emf:
1.77
Enter 5 th reading of peak emf:
1.79
Enter 6 th reading of peak emf:
1.81
Enter 7 th reading of peak emf:
1.83
Enter 8 th reading of peak emf:
1.84
```

RESULTS

Velocities in order:-

8.055365778435364
7.670555311312658
7.218101282864951
6.798435127358322
6.283185307179585
5.8357758271017826
5.076532566397428
4.399429559451014



Experiment 2- DETERMINE THE SPECIFIC ROTATION OF SUGAR USING LAURENT'S HALF POLARIMETER.

The LAURENT'S HALF POLARIMETER is an instrument used for finding the rotation of plane of polarised light, when it passes through the optical active transparent solutions, such as sugar or Rochelle salt solution. When used for finding the optical rotation of sugar it is called Saccharimeter. If the specific rotation is known the concentration of the sugar solution can be determined.
If theta is the optical rotation produced by one decimetres of solution and C is concentration in grams per cc, the specific rotation S at a given temperature And corresponding to a wave length lambda is given by
[s]*λ=Rotation in degrees of optically active solution/length of tube in decimetre x concentration in grams per cc
We use lists and loops to calculate the value of specific rotation
We take input of value of length of tube, concentration of sugar solution; readings of angles for water and sugar solution.

PROGRAM SCREEN:-

```
C:\Program Files\WindowsAp
Enter number of readings taken for each solution(sugar and water): 2

Enter reading of 1 th angle for water solution
80
Enter reading of 2 th angle for water solution
260

Enter reading of 1 th angle for sugar solution
102
Enter reading of 2 th angle for sugar solution
282
Enter the length of the tube IN DECIMETERS: 2
Enter concentration of sugar solution IN GRAM PER CC: 0.2
The value of specific rotation for sugar solution is:  55.0
```

Experiment 3- TO VERIFY THE LAW OF MALUS FOR PLANE POLARISED LIGHT

Malus' Law is a physical principle that describes the relationship between the intensity of plane-polarized light passing through a polarizing filter and the angle between the plane of polarization of the light and the plane of the polarizing filter. According to the law, the intensity of the light passing through the filter is directly proportional to the cosine of the angle θ between the plane of transmission of analyzer and polariser .

I = I' cos^2 (θ)

The principle of Malus' Law can be understood by considering the way that light waves are polarized. When light is unpolarized, it vibrates in all directions perpendicular to its direction of travel. However, when it passes through a polarizing filter, only light waves vibrating in a specific plane are allowed to pass through. This results in the light becoming plane-polarized.

The angle between the plane of polarization of the light and the plane of the polarizing filter determines how much of the light is allowed to pass through. If the two planes are perfectly aligned (the angle is 0 degrees), all of the light will pass through the filter, resulting in the maximum intensity. If the planes are perpendicular (the angle is 90 degrees), none of the light will pass through the filter, resulting in the minimum intensity.

To verify the validity of Malus' Law, we can perform an experiment in which we measure the intensity of plane-polarized light passing through a polarizing filter at different angles of rotation. To do this, we can set up a light source (such as a lamp or a laser) and place a polarizing filter in front of it. We can then rotate the polarizing filter using a rotational stage or a mounting that allows for precise angular adjustments. As we rotate the filter, we can measure the intensity of the light passing through it using a photodetector or a light meter.

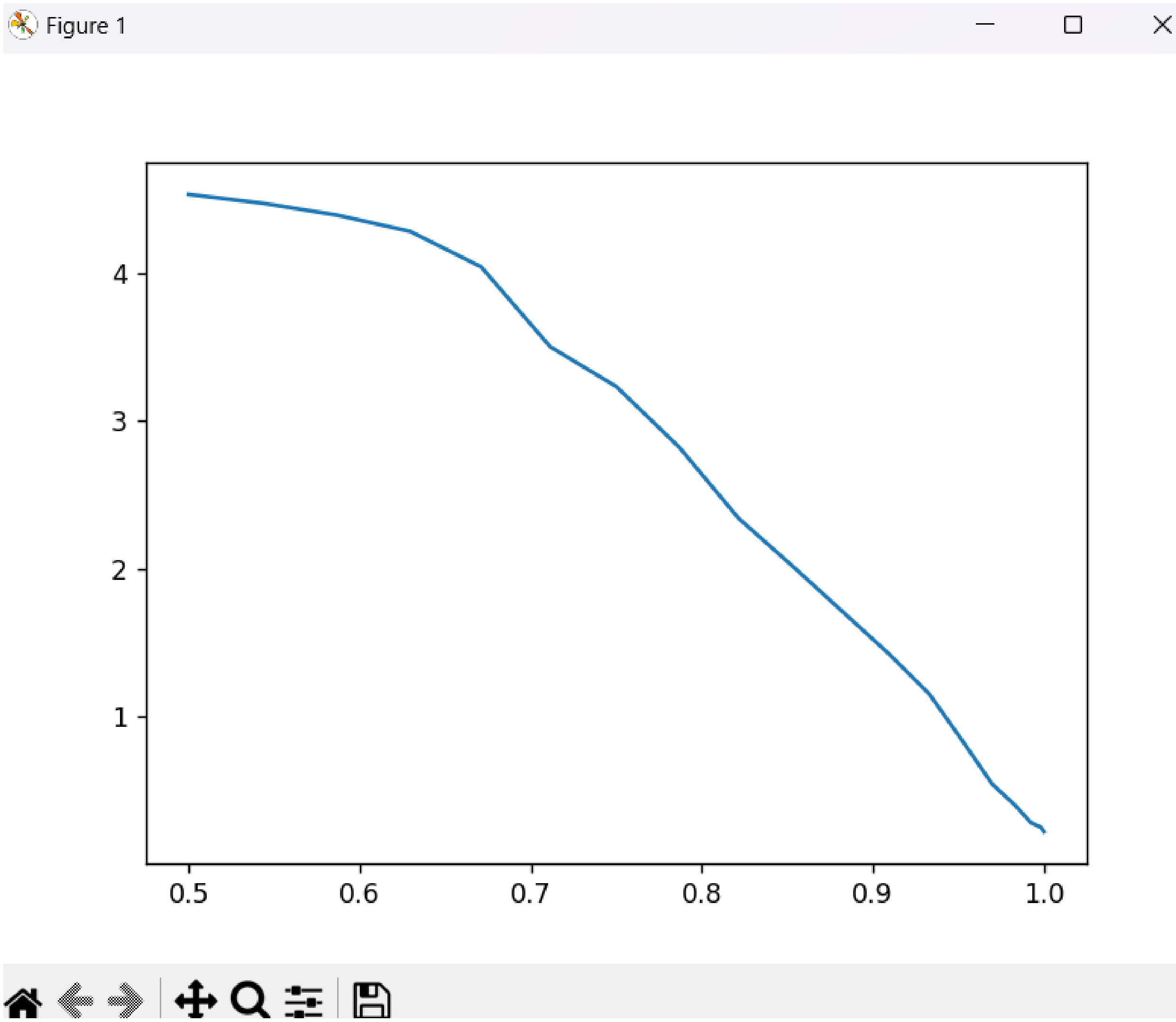
By repeating this measurement at different angles of rotation and plotting the results, we should see a cosine curve, with the intensity of the light reaching a maximum at 0 degrees and a minimum at 90 degrees. This will confirm that Malus' Law holds true for plane-polarized light passing through a polarizing .In this code we use math module to calculate value of cos^2 (θ) , we take current and angle as input from the user and we plot the graph between them by numpy module.

INPUT SCREEN:-

```
Enter the number of readings:19
Enter the 1 th reading of current(CURRENT IS THE INTENSITY):
4.53
Enter the 2 th reading of current(CURRENT IS THE INTENSITY):
4.47
Enter the 3 th reading of current(CURRENT IS THE INTENSITY):
4.39
Enter the 4 th reading of current(CURRENT IS THE INTENSITY):
4.28
Enter the 5 th reading of current(CURRENT IS THE INTENSITY):
4.04
Enter the 6 th reading of current(CURRENT IS THE INTENSITY):
3.50
Enter the 7 th reading of current(CURRENT IS THE INTENSITY):
3.23
Enter the 8 th reading of current(CURRENT IS THE INTENSITY):
2.82
Enter the 9 th reading of current(CURRENT IS THE INTENSITY):
2.34
Enter the 10 th reading of current(CURRENT IS THE INTENSITY):
2.01
Enter the 11 th reading of current(CURRENT IS THE INTENSITY):
1.70
Enter the 12 th reading of current(CURRENT IS THE INTENSITY):
1.42
Enter the 13 th reading of current(CURRENT IS THE INTENSITY):
1.15
Enter the 14 th reading of current(CURRENT IS THE INTENSITY):
0.82
Enter the 15 th reading of current(CURRENT IS THE INTENSITY):
0.54
Enter the 16 th reading of current(CURRENT IS THE INTENSITY):
0.4
Enter the 17 th reading of current(CURRENT IS THE INTENSITY):
0.28
Enter the 18 th reading of current(CURRENT IS THE INTENSITY):
0.25
Enter the 19 th reading of current(CURRENT IS THE INTENSITY):
0.22
```

```
Enter the 1 th reading of angle:
90
Enter the 2 th reading of angle:
85
Enter the 3 th reading of angle:
80
Enter the 4 th reading of angle:
75
Enter the 5 th reading of angle:
70
Enter the 6 th reading of angle:
65
Enter the 7 th reading of angle:
60
Enter the 8 th reading of angle:
55
Enter the 9 th reading of angle:
50
Enter the 10 th reading of angle:
45
Enter the 11 th reading of angle:
40
Enter the 12 th reading of angle:
35
Enter the 13 th reading of angle:
30
Enter the 14 th reading of angle:
25
Enter the 15 th reading of angle:
20
Enter the 16 th reading of angle:
15
Enter the 17 th reading of angle:
10
Enter the 18 th reading of angle:
5
Enter the 19 th reading of angle:
0
```

OUTPUT:-



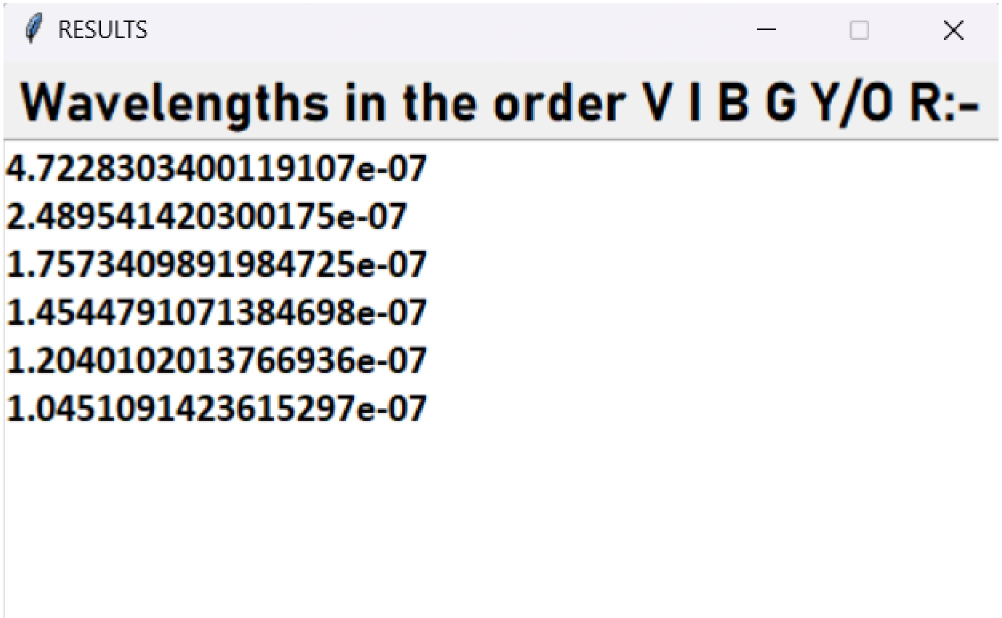
Experiment 4- SPECTROMETER GRATING

Broadly diffraction is defined as deviations is the expectations of ray optics. A diffracted image thus is formed due to deviation of light from the straight-line path. The condition to obtain maximum intensity is that the path difference at a given point is an integral multiple of lambda.
To determine the wavelength the readings are taken by coinciding the prominent lines that could be seen with a vertical wire. The reading are tabulated and from this angle of diffraction for different colours are determined. the values are tabulated on a table. The wavelengths for different lines are calculated using the given formula.
The input taken was the readings of angles calculated from both scales of the apparatus, and the math module was used to perform the calculation of sinx and for the value of pi. The tkinter module was used to display the output using GUI.

INPUT:-

```
C:\Program Files\WindowsAp
Enter the number of lines in grating PER INCH: 15000
Enter the readings taken on scale 1:-
Enter reading for the  Zero Order  fringe:
87.01
Enter reading for the  Violet  fringe:
103.3
Enter reading for the  Indigo  fringe:
104.11
Enter reading for the  Blue  fringe:
105.13
Enter reading for the  Green  fringe:
107.10
Enter reading for the  Yellow/Orange  fringe:
107.83
Enter reading for the  Red  fringe:
108.73

Enter the readings taken on scale 2:-
Enter reading for the  Zero Order  fringe:
267
Enter reading for the  Violet  fringe:
283.10
Enter reading for the  Indigo  fringe:
284.10
Enter reading for the  Blue  fringe:
285.16
Enter reading for the  Green  fringe:
287.10
Enter reading for the  Yellow/Orange  fringe:
287.83
Enter reading for the  Red  fringe:
288.75
```

OUTPUT:-


RESULTS
Wavelengths in the order V I B G Y/O R:-
4.7228303400119107e-07
2.489541420300175e-07
1.7573409891984725e-07
1.4544791071384698e-07
1.2040102013766936e-07
1.0451091423615297e-07

CODING

1.main.py

```
import matplotlib.pyplot as plot
import numpy as np
import math

from tkinter import *
from formulas import *

m=Tk()
m.title("PHYSICS EXPERIMENT CALCULATOR")
m.geometry("1300x500")

label=Label(m,text="SELECT ONE OF THE EXPERIMENTS SHOWN BELOW TO
CONTINUE",font="Arial 18 bold")
label.grid(row=0,column=5,padx=10,pady=10,sticky="N")
label.config(bg="#4FB9AF")

btnf=Button(m,text="FARADAY'S LAW",command=farawin,font="Calibri 14 bold")
btnf.grid(row=1,column=5,padx=10,pady=10)
btnf.config(bg="#4FB9AF")

btnm=Button(m,text="MALUS LAW",command=malus,font="Calibri 14 bold")
btnm.grid(row=1,column=6,padx=10,pady=10)
btnm.config(bg="#4FB9AF")

btns=Button(m,text="SPECIFIC ROTATION",command=sugar,font="Calibri 14 bold")
btns.grid(row=2,column=5,padx=10,pady=10)
btns.config(bg="#4FB9AF")

btng=Button(m,text="SPECTROMETER GRATING",command=grating,font="Calibri 14 bold")
btng.grid(row=2,column=6,padx=10,pady=10)
btng.config(bg="#4FB9AF")

m.config(bg="#008080",border=10)
m.mainloop()
```

2. formulas.py: self made functions stored in this module

```
import math
# import os
from subprocess import call

from tkinter import *
def dist():
    call(["python", "faraday_dist.py"])
def ang():
    call(["python", "faraday_angle.py"])
def farawin():
    f=Tk()
    f.geometry("500x500")
    f.title("Faraday's Law Experiment")
    b1=Button(f,text="Constant Distance",command=dist)
    b1.pack()
    b2=Button(f,text="Constant Angle",command=ang)
    b2.pack()

def malus():
    call(["python", "malus.py"])
def sugar():
    call(["python", "sugar.py"])
def grating():
    call(["python", "grating.py"])
```

3. faraday_angle.py

```
import matplotlib.pyplot as plt
import math
import numpy as np
from tkinter import *
print("FARADAY'S LAW(CONSTANT ANGLE)")
n=int(input("Enter number of readings:"))
emf_list=[]
time_list=[]
dist_list=[]
a=int(input("Enter the constant angle taken:"))
a2=(a/2)*(math.pi/180)
print()
print("Enter the readings of time below:-")
for i in range(0,n):
    print("Enter",i+1,"th reading of time:")
    t=float(input())
    time_list.append(t)
print()
print("Enter the readings of distance below:-")
for i in range(0,n):
    print("Enter",i+1,"th reading of distance:")
    d=float(input())
    dist_list.append(d)
print()
print("Enter the readings of peak emf below:-")
for i in range(0,n):
    print("Enter",i+1,"th reading of peak emf:")
    e=float(input())
    emf_list.append(e)

print()
vel_list=[]
for i in range(0,n):
    v=((4*math.pi)/time_list[i])*(dist_list[i])*math.sin(a2)
    vel_list.append(v)

win=Tk()
win.geometry("500x500")
win.resizable(0,0)
win.title("RESULTS")
Label(win,text="Velocities in order:-",font="Bahnschrift 20 bold").pack()
```

```
text=Text(win,width=80,height=15, font="Calibri 15 bold")
text.pack()
for i in vel_list:
    text.insert(END,str(i)+'\n')
win.mainloop()



xpoints = np.array(vel_list)
ypoints = np.array(emf_list)

plt.plot(xpoints, ypoints)
plt.show()
```

4. faraday_dist.py

```
import matplotlib.pyplot as plt
# import matplotlib
import math
# import sys
import numpy as np
from tkinter import *
print("FARADAY'S LAW(CONSTANT DISTANCE)")
n=int(input("Enter number of readings:"))
emf_list=[]
time_list=[]
angle_list=[]
d=int(input("Enter the constant distance taken:"))
print()
print("Enter the readings of time below:-")
for i in range(0,n):
    print("Enter",i+1,"th reading of time:")
    t=float(input())
    time_list.append(t)
print()
print("Enter the readings of angle below:-")
for i in range(0,n):
    print("Enter",i+1,"th reading of angle:")
    a=float(input())
    a2=(a/2)*(math.pi/180)
    angle_list.append(a2)
print()
print("Enter the readings of peak emf below:-")
for i in range(0,n):
    print("Enter",i+1,"th reading of peak emf:")
    e=float(input())
    emf_list.append(e)

print()
vel_list=[]
for i in range(0,n):
    v=((4*math.pi)/time_list[i])*d*math.sin(angle_list[i])
    vel_list.append(v)

win=Tk()
win.geometry("500x500")
win.resizable(0,0)
```

```
win.title("RESULTS")
Label(win,text="Velocities in order:-",font="Bahnschrift 20 bold").pack()
text=Text(win,width=80,height=15, font="Calibri 15 bold")
text.pack()
for i in vel_list:
    text.insert(END,str(i)+'\n')
win.mainloop()



xpoints = np.array(vel_list)
ypoints = np.array(emf_list)

plt.plot(xpoints, ypoints)
plt.show()
```

5. sugar.py

```
print("***************************SPECIFIC ROATION OF SUGAR SOLUTION***************************")
print()
print()
n=int(input("Enter number of readings taken for each solution(sugar and water): "))
water=[]
ssugar=[]
print()
for i in range(0,n):
    print("Enter reading of",i+1,"th angle for water solution")
    w=float(input())
    water.append(w)
print()
for i in range(0,n):
    print("Enter reading of",i+1,"th angle for sugar solution")
    s=float(input())
    ssugar.append(s)

sum=0
for i in range(0,n):
    sum+=(ssugar[i]-water[i])

avg=sum/n

L=float(input("Enter the length of the tube IN DECIMETERS: "))    #was 2dm in my case
C=float(input("Enter concentration of sugar solution IN GRAM PER CC: ")) #was 0.2 in my case
S=(avg)/(L*C)
print("The value of specific rotation for sugar solution is: ",S)
```

6. malus.py

```
import math
import matplotlib.pyplot as plt
import numpy as np
print("**************************MALUS LAW**************************")
n=int(input("Enter the number of readings:"))
clist=[]
alist=[]
coslist=[]
for i in range(0,n):
    print("Enter the",i+1,"th reading of current(CURRENT IS THE INTENSITY):")
    c=float(input())
    clist.append(c)
print()

for i in range(0,n):
    print("Enter the",i+1,"th reading of angle:")
    t=int(input())
    alist.append(t)
print()

for i in alist:
    a=(i/2)*(math.pi/180)
    cos=(math.cos(a))**2
    coslist.append(cos)

xpoints = np.array(coslist)
ypoints = np.array(clist)

plt.plot(xpoints, ypoints)
plt.show()
```

7. grating.py

```
import math
from tkinter import *
m=float(input("Enter the number of lines in grating PER INCH: "))
order=["Zero Order","Violet","Indigo","Blue","Green","Yellow/Orange","Red"]
sc1=[]
sc2=[]
print("Enter the readings taken on scale 1:-")
for i in range(0,7):
    print("Enter reading for the ",order[i]," fringe:")
    t1=float(input())
    sc1.append(t1)
print()
print("Enter the readings taken on scale 2:-")
for i in range(0,7):
    print("Enter reading for the ",order[i]," fringe:")
    t2=float(input())
    sc2.append(t2)
print()
sum1=0
sum2=0
a1=[]
a2=[]
for i in range(0,6):
    sum1+=(sc1[i+1]-sc1[0])
    a1.append(sum1)
    sum1=0
    sum2+=(sc2[i+1]-sc2[0])
    a2.append(sum2)
    sum2=0
avg=0
tavg=[]
for i in range(0,6):
    avg=(a1[i]+a2[i])/2
    tavg.append(avg)
lines=m/0.0254
wave=[]
for i in range(1,7):
    ang=(tavg[i-1])*(math.pi/180)
    w=(math.sin(ang))/(lines*i)
    wave.append(w)
win=Tk()
```

```python
win.geometry("500x500")
win.resizable(0,0)
win.title("RESULTS")
Label(win,text="Wavelengths in the order V I B G Y/O R:-",font="Bahnschrift 20 bold").pack()
text=Text(win,width=80,height=15, font="Calibri 15 bold")
text.pack()
for i in wave:
    text.insert(END,str(i)+'\n')
win.mainloop()
```

LIMITATIONS

- Files have to be pasted in the root directory
- The pop up windows which appear after clicking on an option don’t show up over existing window
- Some readings have to be given in specific units

REQUIREMENTS

- <u>Hardware Configuration:</u>

- *Processor :* Intel Pentium Or Equivalent And Above
- *Memory :* 2 GB Ram
- *Hard disk :* 1 GB Free Space

- <u>Software Configuration:</u>

- *Platform :* Windows 7 Or Later
- *Software:* Python 3 Or Later