LAB 1: INTERFERENCE / DIFFRACTION WITH PYTHON

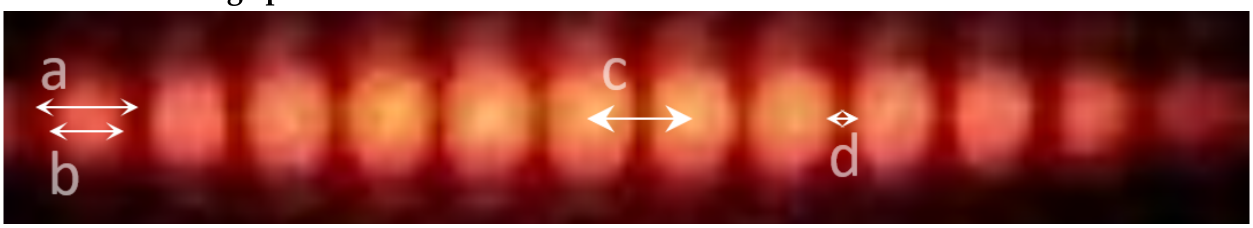
OBJECTIVE:

Investigate interference / diffraction by eye and using a python-controlled photodetector

PRELAB[1]:

Those of you who have taken PHYS 195 (and probably more of you) have previously learned about slit diffraction, but you may wish to review by reading your previous physics textbook (probably Halliday, Resnick, and Walker, Chapters 35 and 36.) If your physics background has not prepared you for this, then come to office hours or make an appointment with your lab instructor.

Examine the fringe pattern below:



1. A lab team wants to measure a fringe pattern from a double-slit and whether that corresponds to the slit width and spacing for this double slit. What's either ok or incorrect about these four examples **a**, **b**, **c**, and **d** (see above) of measuring the distance between fringes?
2. Which feature of the double-slit mask is mathematically related to the measurement of the fringe pattern?
3. What sort of measurement would be more precise than *any* of these four examples?

Coding practice:

To practice some of the structures and functions you'll need in your code for this lab, complete the exercises below.

1. Write a pseudocode that uses a for loop to create a 100-element array of x-values from -1 to 1 and a 100-element array of y-values for $y=x^2$ and plot y vs x.
2. Turn your pseudocode into a working python code. Here are some things that will be useful:

Modules you need to import	numpy to create arrays: import numpy as np matplotlib.pyplot for simple plots: import matplotlib.pyplot as plt

[1] Photo credit: https://commons.wikimedia.org/wiki/File:Single_slit_and_double_slit2.jpg and I (Viva) added the dimension bars to the image.

Creating an array	There are many kinds of arrays. Check out Numpy's user guide for examples. Megan's favorite is zeros
Plotting in python	For the plots you'll be making today, you'll use matplotlib.pyplot's plot function. It takes two arguments (your x and y arrays) and plots them: plt.plot(x,y) to make a y vs x plot

INTRODUCTION

When coherent light passes through a mask, the interference pattern created depends precisely on the shape of the mask. We can predict the mask shape from the interference pattern and the interference pattern from the mask. We will measure this interference pattern both by hand and by using a photodetector mounted in a motorized translation stage. When controlled by the appropriate commands, this translation stage will move the photodetector, which will measure the intensity of the light as it moves. By saving the intensity data, you will be able to plot the intensity vs position graph and measure the interference pattern much more precisely than your previous measurements.

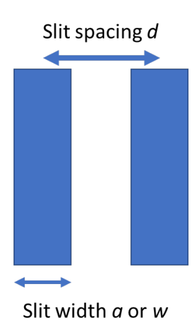
Photodetector and translation stage

APPARATUS

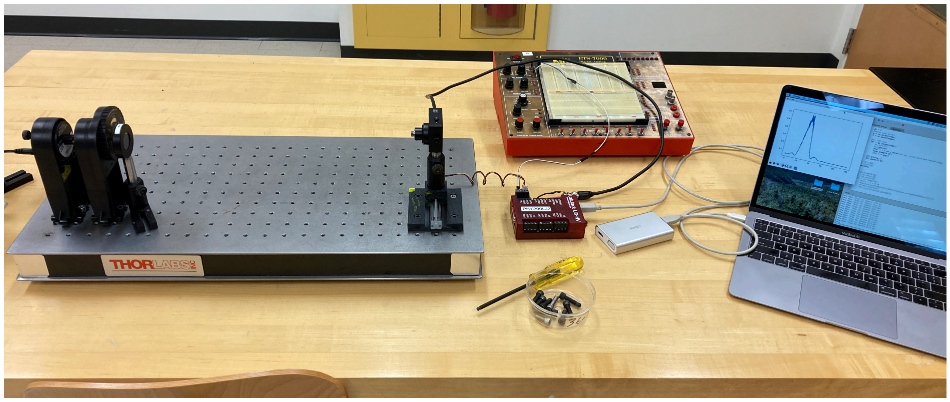
We have given you an optical bench with a laser, a viewing screen, and two slit wheels, each with a bunch of different slit patterns that can be selected by rotating the wheel. For the second portion of the lab, we have provided a photodetector mounted in a translation stage. These communicate with your computer via a LabJack and are powered by the ETS-7000 system.

PROCEDURE

1. Place your laser and single slit wheel on the breadboard so that you can project the interference/diffraction pattern of a single slit onto the wall. Trace this pattern into your lab notebook. If you would like, there are paper strips with rulers printed on them for ease of measurement.
2. Replace the single slit wheel with the double slit wheel and trace this pattern. Take the appropriate measurements to verify **slit spacing *d*** with the value printed on the wheel.



3. Place the motorized translation stage and photodetector in a position where the photodetector will scan across the fringe pattern. Your set up will look something like the figure below.



4. Write a **python code** that drives the motor of the translation stage using the LDrive.py code provided. Test that you can move the photodetector in both directions and try different numbers of steps. See Information below for a list of the functions in LDrive.py. *Note*: If you have both a USB-A port and python on your computer, you are welcome to use your own computer. You may alternatively log in to the lab computer with username: pguest and password: physics is phantastic, and open the software "Spyder" through the "Anaconda Navigator".
5. **Calibrate** the relationship between the number of steps used in your code and the distance the photodetector travels by moving the translation stage 200 steps and measuring the distance traveled.
6. Add to your code so that it drives the motor, measures the intensity of the interference pattern, stores this value and the position of the photodetector to numpy arrays, and saves your arrays to use later in the semester.
7. Produce a graph of intensity vs position for the diffraction pattern you measured, with two versions: one version has the intensity on a linear scale and the other has it on a logarithmic scale. Describe all features of the graph. If the signal saturates, you need to use an optical density (OD) filter to your optical apparatus before the photodetector.

LAB ASSIGNMENT

Write a methods section for this lab.

INFORMATION

Software: Writing python script

Your task is to write a python script that moves the photodetector mounted on a translation stage. Your program should move the stage a certain number of steps, then stop to record the light signal at that position before moving on.

Start by making a list of all the things you want your code to do. Think about the order of each task. Whatever needs to be repeated should go in a **for loop**. Anything that does not need to be repeated should stay outside the for loop.

To complete today's lab, your code needs to import a few **modules**:

```
import matplotlib.pyplot as plt
import numpy as np
import time
import u3
```

Check to see if that runs before you go on. If it has an error that says "no module named matplotlib" then your instructor can help you.

Additionally, you need to import **LDrive**, the module written by Brian. First, download this module from the Google Drive (), then open it and import with

```
from LDrive import LDrive
s = LDrive()
```

Calling LDrive() creates an LDrive object, which we've named ***s*** (for scan). You have to store the object in some variable name but it doesn't have to be called s. Doing this gives you access to a few very helpful functions that communicate with the LabJack. You are welcome to read the LDrive.py code and make sense of these functions for yourself, but here's the summary:

Functions provided by LDrive.py:

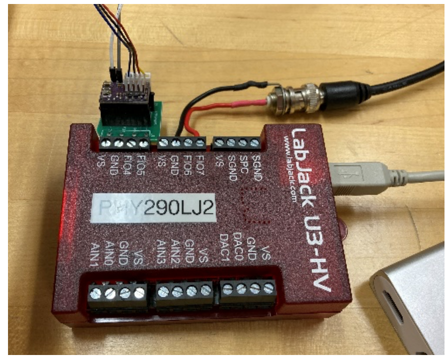
Function Name	**Description**
s.left(n) s.right(n)	The left and right functions drive the translation stage forward and backward n-steps. We don't know the physical distance of 1-step. You need to determine that through calibration.
s.scan(m)	This function reads the input signal from the photodetector m times in a row. It produces two arrays: time and signal. The function s.avg() uses this.
s.avg()	This function runs s.scan(100) and then averages those 100 points to produce a single output. This is the function you will call in your code.

You may also use the if-then-else structure:

```
if _____:
   [run commands that are indented]
else:
   [run commands that are indented]
```

You may read more about if statements here:
https://www.w3schools.com/python/python_conditions.asp

Automation Hardware Connections



The LabJack is connected for you this week, but here's a description of how that is done.

A circuit board has been installed on the LabJack at pins FIO5, FIO4 (IO=input and output), GND (ground), VS (voltage supply). This controls the motorized translation stage. The translation stage connector plugs to the four-pin input. A 15V potential difference is required to drive the motor; this is connected to the two-pin input (the +15V wire should be attached to the bottom (outside) pin).

The photodetector input is connected to FIO7 and GND.