Experiment on Diffraction and Interference Phenomena

Filippo Romiti

Prof. —

Abstract—This experiment deals with diffraction and interference phenomena, which we replicated in the laboratory on 25/3, 27/3 and 18/4. The following analysis involves estimating the wavelength of a red light laser (650nm) by obtaining the data manually and through sensors, and then proceeding to compare the brightness patterns produced by both phenomena with the respective mathematical models.

keywords— *wave, diffraction, interference*

Contents

1. Introduction

Diffraction and interference phenomena are governed by a very precise and relatively simple physical-mathematical model. We performed the following experiments in order to understand what the studied theory meant on a practical level. The first objective will be to calculate the wavelength of the laser (coherent and monochromatic) from data taken by hand and through luminosity sensors. Then we will proceed by comparing graphs of both phenomena with their respective models in order to find similarities and differences.

2. Theory

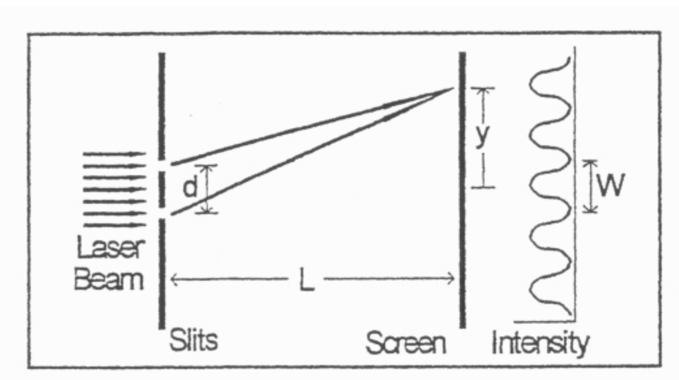

Figure 1. schematic image of the experiment (interference - double slit)

2.1. Nomenclature

d - distance between slits
a - amplitude of a slit
θ - angle on the laser beam
L - distance slit-screen
y - distance centre-incidence point
k - index of max/min from center
λ - wavelength

2.2. Applications

diffraction and interference are two distinct phenomena. both are governed by a similar law, which takes two different meanings. In order to witness diffraction phenomena only one slit is needed, while for interference ones two. For this reason, one can experience clean diffraction phenomena, but not interference phenomena without an imposed diffraction. In fact, in addition to producing an interference in the presence of two or more slits, each of these will also produce its own diffraction.
The formula behind these phenomena is:

$$d \sin\theta = k\lambda \tag{1}$$
$$a \sin\theta = k\lambda \tag{2}$$

For every k, this formula represents a local maximum of luminosity (interference) or a minimum (diffraction), with the absolute maximum always being located in the middle ($\theta = 0$)

As in the figure, $d \mid a$ represents the distance between slits for interference phenomena or the amplitude of a slit in case of diffraction. In presence of small angles (especially in diffraction), the sine of the angle is approximately the $y\,/\,L$ ratio, since:

$$\sin\theta = \tan\theta = \frac{y}{L} \tag{3}$$

This is everything needed to calculate wavelength from experimental data. The next part of the experiment involves comparing the graphs of light intensity in the cases of diffraction and interference (with superimposed diffraction) with the corresponding mathematical models.
The diffraction one is pretty simple, given by this formula:

$$I = I_0 \left(\frac{\sin^2 \phi}{\phi^2} \right) \tag{4}$$

with ϕ being:

$$\phi = \frac{\pi a \sin\theta}{\lambda} \tag{5}$$

Interference one is quite more complex, since it contains a part concerning the interference itself but also one calculating the diffraction produced by the two single slits.

$$I = I_0 \cos^2 \left(\frac{\pi d \sin\theta}{\lambda} \right) \left(\frac{\sin \frac{\pi a \sin\theta}{\lambda}}{\frac{\pi a \sin\theta}{\lambda}} \right)^2 \tag{6}$$

The first part, considering d (distance between slits) is the actual interference one. The second one calculates the effect of diffraction from the singularly considered slits. In fact, it uses a (amplitude of a slit) as parameter.

3. Experimental Setup

The experimental setup involves the following tools:

- White Screen

- Wireless Luminosity Scanner
- Diffraction and Interference Slits
- Red Laser
- Optic Metallic Bar, 1.2 m

The white screen and the scanner where used respectively for manual measures and digital data capture.

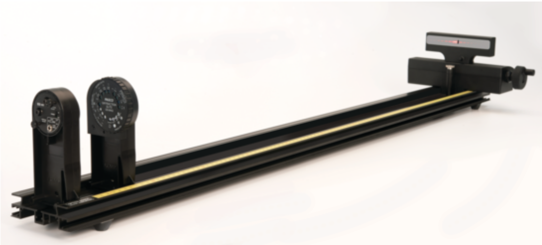

Figure 2. experimental setup - PASCO scientific

Data was captured trough PASCO Capstone Software, exported as comma separated value files and later analyzed in Python.

4. Constant Data

During the whole experiments, these were the constant parameters:
L = 0.9 m
a (diffraction) = 0.16 mm
d (interference) = 0.5 mm
a (interference) = 0.08 mm

5. Wavelength Estimation

We can calculate wavelength in both experiments, with hand measurements and digital captured data analysis. To calculate the wavelength, we used equation 1 for interference and equation 2 for diffraction in their inverse form combined with equation 3.

$$\lambda = \frac{\gamma y}{kL} \tag{7}$$

In which γ stands for a or d

5.1. Manual Measures

Next step consists in placing a sheet of paper on the white screen, on which the pattern created by the slit is projected. This way it's possible to measure the patterns with a pencil and a ruler (the y data), in order to make an estimation of the wavelength.

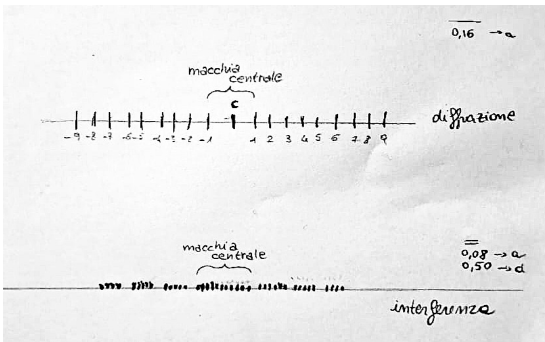

Figure 3. sheet of paper with manual taken data

5.1.1. Diffraction

Considering the amplitude $a = 0.16$ mm and the second minimum of the pattern, being at $y = \pm 7.5$ mm with $k = 2$, we got:

$$\lambda = \frac{ay}{kL} = \frac{0.16 \times 10^{-3} \times 7.5 \times 10^{-3}}{2 \times 0.9} = 6.66 \times 10^{-7} \tag{8}$$

since the declared wavelength is 650 nm, the result we got from this measure is pretty much correct (error of 2.46%). Testing with $k = 3$ and $y = 11.0$ mm will probably still confirm the result:

$$\lambda = \frac{ay}{kL} = \frac{0.16 \times 10^{-3} \times 11.0 \times 10^{-3}}{3 \times 0.9} = 6.51 \times 10^{-7} \tag{9}$$

With this manual approach, it's easier and more reliable to measure longer distances for higher values ok k, since measurements with rulers aren't precise enough. This resulted in this case in an error of 0.15%

5.1.2. Interference

The only differences with interference are that the value of $d = 0.5$ mm is the one that counts and the relative maximums (which are much closer than in the diffraction) are the ones represented by integer multiples of k. For $k = 9$ the spikes are located at $y = \pm 11$ mm:

$$\lambda = \frac{dy}{kL} = \frac{0.5 \times 10^{-3} \times 11 \times 10^{-3}}{9 \times 0.9} = 6.79 \times 10^{-7} \tag{10}$$

Manual measurements on interference resulted in a worse result (with an error of 4.46%) than the ones on diffraction, probably since the very close maximums make it harder to obtain regular and precise data to base the estimation on. The wide range of errors obtained suggests that manual data extrapolation in an inconsistent method to apply.

5.2. Sensor Assisted measures

The same data are taken with a mobile brightness sensor, connected to the computer via Bluetooth. on the device, the sensor exposure can be adjusted, and other adjustments such as sampling rate (frequency) can occur via software. When recording, the software compiles a csv file containing the following parameters:

- time (every f^{-1} seconds)
- sensor position
- light intensity detected
- logarithm of the intensity

By sliding the sensor longitudinally with respect to the laser beam, it records for each sampling fraction all these data. This produces a printout containing the pattern projected on the sensor screen.

By having access to the entirety of the data, more precise estimates can be made regarding wavelengths. In fact, by averaging through the data collected at the various k, we can reasonably expect fairly accurate results. Two simple Python functions (one for diffraction and one for interference) were used for this purpose. The parameters to be taken into account are the range over which the analysis is to be applied, the k values in it and the respective y values (max or min). This can be expressed as:

$$\lambda_m = \frac{\sum_{k=-k_{max}}^{k_{max}} \frac{\gamma y_k}{kL}}{|2k_{max}|} \tag{11}$$

In which k_{max} indicates the absolute value of the maximum sampling index on which to operate and γ stands for a o d. λ_m is the medium value of λ across the considered data range. A wider range will always provide a higher accuracy. Here's an example of implementation:

```
def wavelength_interference(dataset):
    wls = []
    for i in dataset:
        wls.append((d*(i[1] - imaxd))/(L*i[0]))
    wl = sum(wls) / len(wls)
    return wl
```

Assuming "dataset" to be a list containing lists of two dimensions, with ordered values of k and their respective y, this function calculates the average wavelength, considering L and γ constant. y is calculated as relative distance from the absolute maximum. Put this way, the function is not the exact equivalent of Equation 11, since it calculates the mean regardless of the order of the data. To follow precisely the equation, data need to be passed in a way like this:

```
data = [
    [1, 77.08e-3],
    [2, 78.27e-3],
    [3, 79.45e-3],
    [-1, 74.76e-3],
    [-2, 73.58e-3],
    [-3, 72.48e-3]
]
```

With the integers at indexes zeros being the k values and the floats at indexes ones the respective y values.
This is always data extracted by hand from the csv file and copied into the python file without any automation. It is indeed a complicated operation to extrapolate the positions of the relative minima, especially in the case of interference with superimposed diffraction.

5.2.1. Diffraction

Diffraction data is passed to the function, calculating the medium wavelength result and its error in percentage.

```
print(wavelength_diffraction(ddata),
(650e-9 - wl_diffraction(ddata)/(650e-9))*100)

>>> 6.497962962962964e-07, 0.03133903133902223
```

With k values ranging from -4 to +4, the result is exact, with only 0.03% of error on the declared 650 nm.

5.2.2. Interference

The same is applied on interference data.

```
print(wavelength_interference(idata),
((650e-9 - wl_interference(idata))/(650e-9))*100)

>>> 6.470679012345684e-07, 0.451092117758719
```

k values ranged in this calculation from -3 to +3. The error here is 10 times higher, with a 0.4%. This is probably caused by the fact that the measuring instrument was better suited to detect differences in brightness in greater ranges of positions than those produced by an interference phenomenon.
The data considered in the calculation trough diffraction range for a total of 30 mm, while those of interference are enclosed in a total range of only 7 mm.

5.3. Considerations on Wavelength Estimation

Collecting data to estimate the wavelength by manual measurements can produce results that are quite close to the real ones, but it is definitely not a reliable and repeatable method to apply. Computer help is definitely an advantage, but care must be taken to properly calibrate the instruments at the time of sampling to avoid insufficient sampling rates, over or under saturation of the sensor.

6. Plot and Comparison with the Models

Having a very dense set of data recorded in the laboratory at frequencies of 50 hz for both experiments, we can expect that a plot of such data can fairly accurately represent the physical phenomenon studied. To verify this we will compare the plots of the experimental data with the ones of theoretical data, derived from equations 4 and 6, in the form of I-functions of variable y.

Datasets have been analysed in two separate .ipynb files. This file format allows the user to choose which part of the code to run, and such feature is very useful when working with data and plots in order to easily spot errors or bugs.

6.1. Diffraction

6.1.1. Experimental

To analyze the experimental data, import the dataset in .csv format into Python and convert it to tabular format using the well known Pandas library. The most relevant columns are those concerning the position and percentage of intensity at that point.

The data cleansing process involves only a normalization of the intensity values between 0 and 100, a relativization of the position data with respect to the Intensity peak, which takes value 0, and a restriction of the domain between two arbitrary values equidistant from the maximum. This whole process can be found commented in the diffraction.ipynb file. Plotting then the cleansed values, the result is the following:

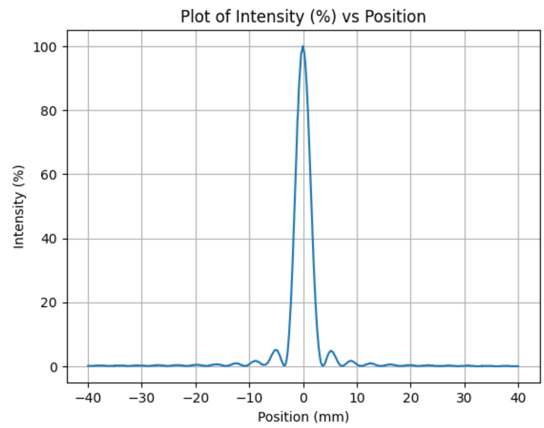

Figure 4. Experimental Diffraction - plot of Intensity in function of Position

It represents a classical diffraction pattern, so we can say that our data were right.

6.1.2. Theoretical

For the theoretical curve, equation 4 is used by transforming it into a function of y. Actually y does not appear explicitly in that equation, but only as the parameter of theta angle.

The diffraction angle with respect to the laser beam we are measuring is quite small: having a distance from the sensor of 0.9 m and a maximum y of 0.075 m, the angle would result in $\arctan \frac{0.9}{0.075} = 4.76°$

As seen in the theory section, the sine of such an angle could be approximated to $\sin\theta = \frac{y}{L}$. Anyway, having a computer and not needing to do calculations by hand, it is preferred to proceed without any approximation (so with $\sin\theta = \sin\arctan\frac{y}{L}$), in order to obtain the most precise result possible.
These considerations justify the following function:

$$I(y) = I_0\left(\frac{\sin^2\phi}{\phi^2}\right) \mid \phi = \frac{\pi a \sin\arctan\frac{y}{L}}{\lambda} \tag{12}$$

This function can be coded as follows in Python:

```
def teorical_intensity(y, L, lmb, a):
  sintheta = math.sin(math.atan((y * 1e-3) / L))
```

```
  phi = (PI * a * sintheta)/lmb
  I = Izero * ((math.sin(phi)**2) / (phi**2))
  return I
```

With the only true variable being y and L, λ and a being constant parameters. Iterating through that function for 0.15 m with a step of 0.0001 m, thus using the same sampling range as the experimental data, we obtain an array composed of subarrays containing an arbitrary value of y (the iterated one) and the respective value of I.

```
teorical_curve = []
lamb = 6.5e-7
L = 0.9
a = 0.16e-3

for i in np.arange(0, 150, 0.1):
  teorical_curve.append(
  [i, teorical_intensity(relx(i), L, lamb, a)]
  )
```

Applying the same data cleansing strategies used for the experimental one, the resulting plot is the following:

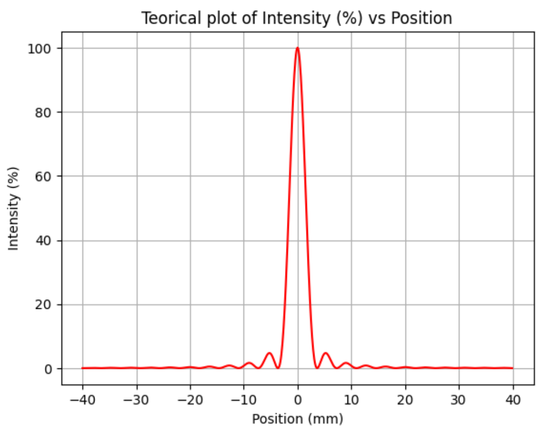

Figure 5. Theoretical Diffraction - plot of Intensity in function of Position

This plot also represents a diffraction phenomenon well, so we can say that we have correctly applied the theoretical model.

6.1.3. Considerations

Plotting the theoretical and experimental graph together, the following result is obtained:

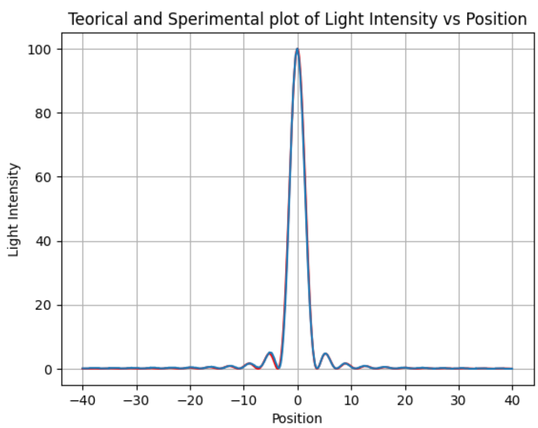

Figure 6. Diffraction Comparison - Experimental (blue) and Theoretical (red)

The result shows that the theoretical graph and the experimental graph coincide perfectly at almost every point. This means that the experimental process was performed correctly. The only stretch where the two curves do not perfectly coincide is between -8 and -3 mm, but the difference between the two is still almost irrelevant.

6.2. Interference

6.2.1. Experimental

The whole same process is applied in the analysis on the experimental interference data. Data is imported from csv and the cleansed in the same way trough pandas and numpy. The resulting plot is the following:

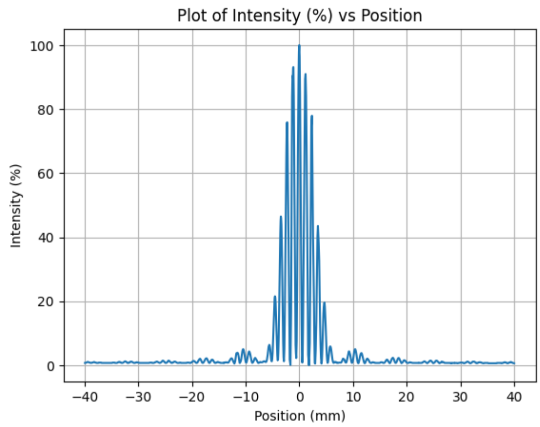

Figure 7. Experimental Interference - plot of Intensity in function of Position

It can be seen that this pattern is exactly as expected, as it shows the dense regular curve typical of interference, adjusted by the inevitable diffraction due to passing through individual slits. Considering that presumably the experimental data were sampled correctly

6.2.2. Theoretical

The theoretical curve is generated and cleansed in the same way as the diffraction curve. The equation considered is the equation 6 transformed as a function of y.

The same considerations regarding angle calculation can be applied here, especially since the interference is especially noticeable considering very small angles. But for the same reason as before we will still proceed by angle calculation without any approximation. Considering this, we can define the following function:

$$I(y) = I_0 \cos^2\left(\frac{\pi d \sin\theta}{\lambda}\right)\left(\frac{\sin\frac{\pi a \sin\theta}{\lambda}}{\frac{\pi a \sin\theta}{\lambda}}\right)^2 \mid \theta = \arctan\frac{y}{L} \tag{13}$$

This function can be coded as follows in Python:

```
def teorical_intensity(y, L, lmb, d, a):
  sintheta = math.sin(math.atan((y * 1e-3) / L))
  I = Izero * (math.cos((PI * d * sintheta) / lmb)**2)
    * (((math.sin((PI * a * sintheta)/ lmb))/
    ((PI * a * sintheta)/ lmb))**2)
  return I
```

As before, the only true variable is y and the array is compiled the same way. Here one more parameter is passed, and it stand for d, which is the distance between the two slits.

```
teorical_curve = []
lamb = 6.5e-7
```

```
fend_distance = 0.5e-3
L = 0.9
a = 0.08e-3

for i in np.arange(0, 150, 0.1):
  teorical_curve.append(
  [i, teorical_intensity(relx(i),
  L, lamb, fend_distance, a)]
  )
```

The graph thus produced results in:

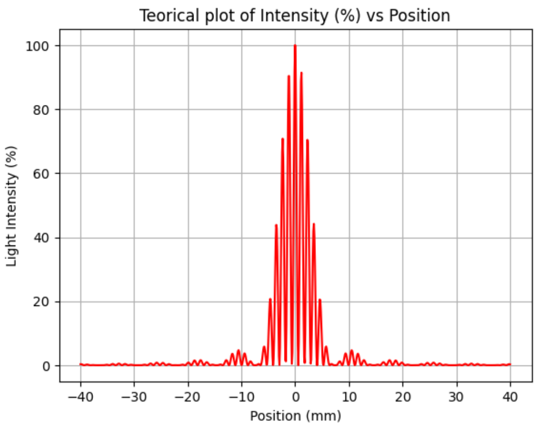

Figure 8. Theoretical Interference - plot of Intensity in function of Position

Once again the graph seems to comply in all points with the governing law, so it has been implemented correctly.

6.2.3. Considerations

The plot of the practical graph superimposed on the theoretical one results in the following:

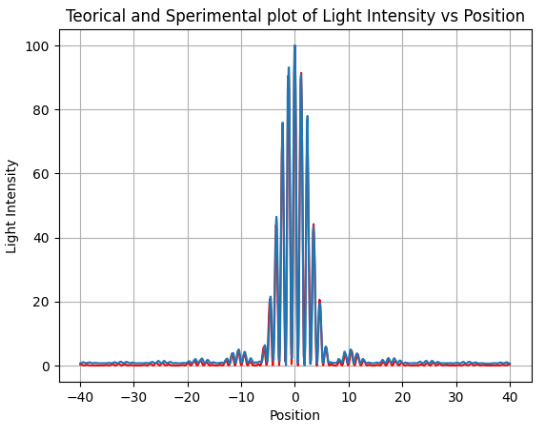

Figure 9. Interference Comparison - Experimental (blue) and Theoretical (red)

Again, the graphs overlap almost perfectly, confirming the correctness of the experimental data. Nevertheless, more errors are noticeable than in the diffraction graph. This is surely due to the fact that the interference presents a much denser graph with very different values placed quite close together. This leads to measurement errors caused both by calibration errors (e.g., of the exposure) and by physical limitations of the instrument (such as the sampling rate, which is limited to 50 hz).

Also noticeable is that values tending to zero are lower in the theoretical curve. This can probably be explained simply by assuming that there was no total darkness in the room at the time of experimental sampling

7. Conclusions

In conclusion, we can say can the process of experimental data collection in the laboratory turned out to be generally successful, as these data are found to be totally in accordance with the theoretical models. This was verified first by calculation of the laser wavelength, which led to good results, and then by plotting and comparison of the curves, which was also successful.

github.com/rmtfpp/int-diff
romiti.filippo@liceocopernico.brescia.it