3rd Year Computational Physics : Jessica Murphy

19/10/2023

Assignment 2

This Python notebook demonstrates how to model the Double Slit Experiment to plot the intensity patterns of light of two slits and one slit, in both 1-D and 2-D. The following formula gives the horizontal variation of the intensity of light at a screen after passing through a double slit:

$$I = I_0\left[\frac{sin(\alpha)}{\alpha}\right]^2[cos(\beta)]^2,$$

$$\alpha = \frac{\pi a}{\lambda}sin(\theta) \text{ and } \beta = \frac{\pi d}{\lambda}sin(\theta)$$

where I_0 is the initial intensity, a = 0.09 mm is the width of one of the slits, L = 480mm is the distance from the slits to the screen, d = 0.4 mm is the distance between the slits, λ = 670 mm is the wavelength of light.

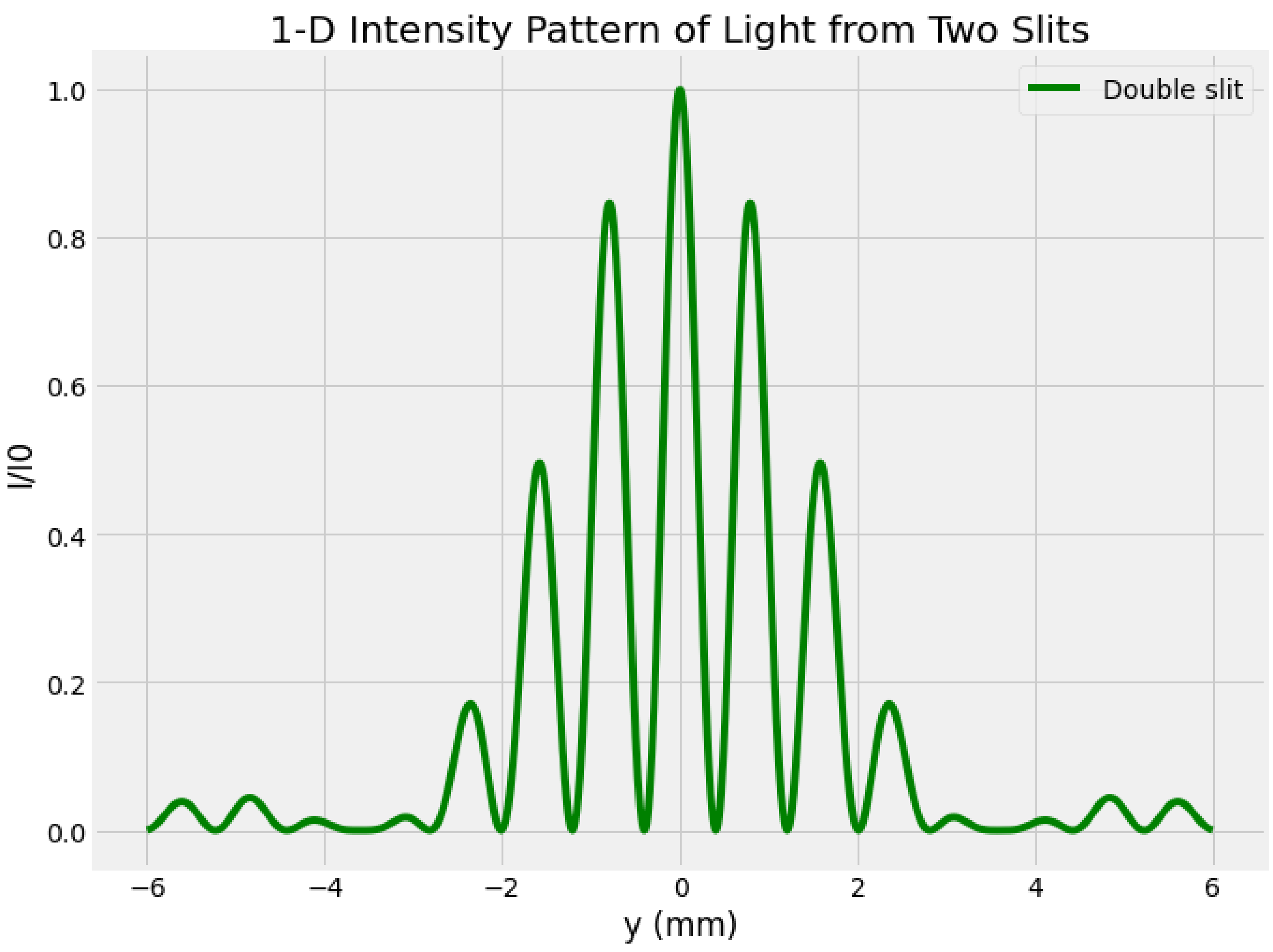
The double slit experiment shows the **wave-particle duality** of matter and the probabilistic nature of quantum mechanics. It was first performed by Thomas Young in 1801 and demonstartes quantum properties such as *superposition* and *interference*.

Task 1

The objective of this task is to plot the predicted 1-D intensity pattern $I/I0$ of light observed from two slits. This should be a pattern of peaks and troughs as a function of y in millimetres, and would be what you would attempt to measure in the lab. Note that $tan(\theta) = \frac{y}{L}$, and the x axis is the direction from the slit to the screen. For the purposes of this notebook, the 'opposite' of this angle is denoted as 'y' to follow the diagram given in the 'Lab 2 slides' pdf.

1. Import the python library NumPy as np, which is the natural log funtion, and maptplotlib.pyplot for the purpose of plotting
2. Set the values of the constants a,L,d,λ
3. Create an array of y values in mm and calculate theta by calculating tan inverse of $\frac{y}{L}$
4. Calculate α,β, and therefore calculate $\frac{I}{I_0}$ for two slits
5. Plot the 1-D intensity of light form two slits as a function of y (mm)

In [10]:
```
# task one
# plotting the predicted 1-D intesnity pattern I/I0 of light observed from two slits

# import python libraries
import numpy as np # import NumPy for the natural log function
import matplotlib.pyplot as plt # import matplotlib to plot graph
# instruction to the kernel to prepare for plot instructions
%matplotlib inline


# set the initial conditions of the parameters
a = 0.09  # Width of one of the slits (millimeters)
L = 480 # Distance from the slits to the screen (millimeters)
d = 0.4  # Distance between the slits (millimeters)
wavelength = 670e-6 # Wavelength of the light in micrometers scaled back by 10^3 so we have graph values in millimetres

# Create an array of y values in millimeters and therefore calculate theta
# note that y is denoted as the 'opposite' of the angle throughout this code as it was described this way in the supplementary diagram for this assignment
y = np.linspace(-6, 6, 1000)  # linearly spaced array for y values
theta = np.arctan(y / L) # theta is equal to tan inverse of x/L

# calculating alpha, beta, and therefore I/I0 for two slits
alpha = (np.pi * a / wavelength) * np.sin(theta)
beta = (np.pi * d / wavelength) * np.sin(theta)
intensity_twoslits = (np.sin(alpha) / alpha)**2 * (np.cos(beta))**2 # initial equation

# Create the plot
plt.figure(figsize=(10, 8)) # increase figure size
plt.style.use('fivethirtyeight') # for background use style sheet from matplotlib
plt.plot(y, intensity_twoslits, 'g-', label='Double slit')
plt.xlabel('y (mm)') # x label
plt.ylabel('I/I0') # y label
plt.title('1-D Intensity Pattern of Light from Two Slits') # title of graph
plt.legend()
#plt.grid()
plt.show()
```



Task 2

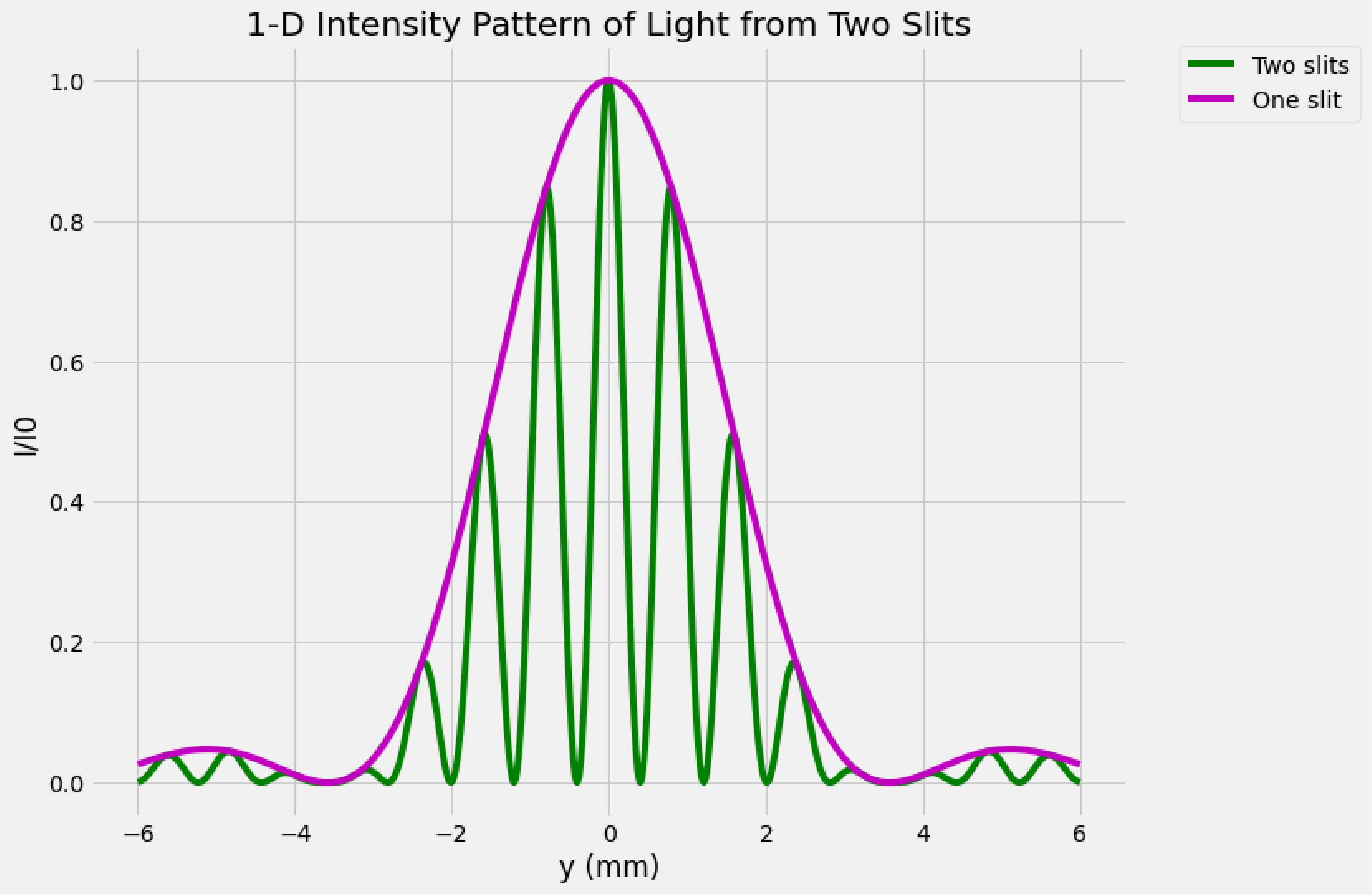
The purpose of this task is to plot the predicted 1-D intensity pattern of light observed from just one of the slits on the same figure. For a single slit, the equation for the intesnity of light reduces to:

$$I = I_0\left[\frac{sin(\alpha)}{\alpha}\right]^2$$

1. Calculate the intensity for one slit
2. Create a plot of this with the intensity of two slits

In [9]:
```
# task two
# plotting the predicted 1-D intesnity pattern I/I0 of light observed from two slits and one slit

# calculating I/I0 for one slit
intensity_oneslit = (np.sin(alpha) / alpha)**2
```

```
# Create the plot
plt.figure(figsize=(10, 8)) # increase figure size
plt.style.use('fivethirtyeight') # for background use style sheet from matplotlib
plt.plot(y, intensity_twoslits, 'g-', label='Two slits')
plt.plot(y, intensity_oneslit,'m-', label='One slit')
plt.xlabel('y (mm)')
plt.ylabel('I/I0')
plt.title('1-D Intensity Pattern of Light from Two Slits')
plt.legend(bbox_to_anchor = (1.05, 1), loc = 'upper left', borderaxespad=0) # adding legend to top right corner
#plt.grid()
plt.show()
```



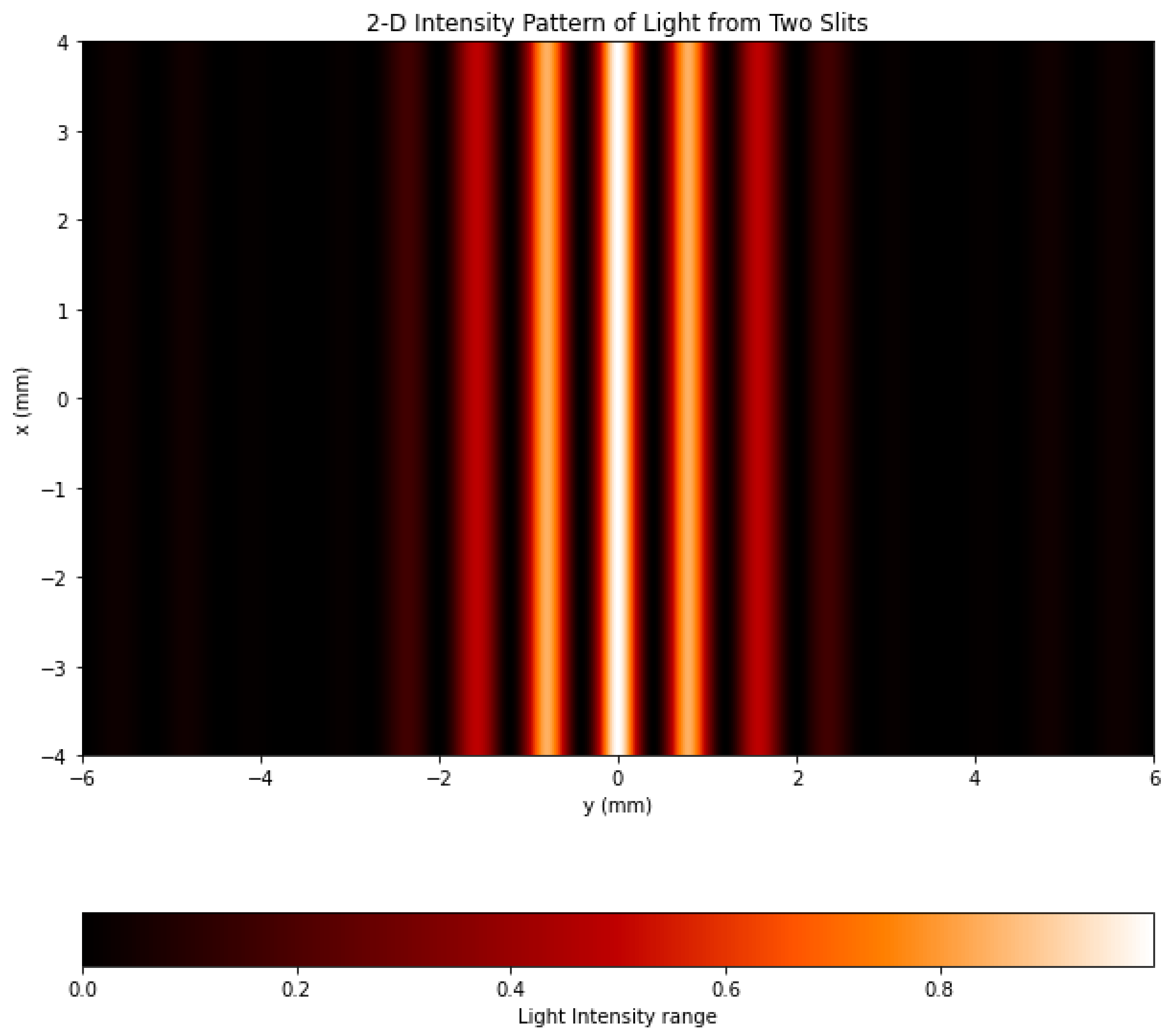
Task 3

For this section of code, the aim was to develop this program to simulate the appearance of the pattern in 2-D on the viewing card in the experiment. Initially here, it is assumed that the 1-D pattern is repeated in the vertical direction.

To achieve this, the existing code is extended to create a 2-d colour plot. It is assumed the 1-D pattern is repeated in the vertical direction.

1. Create an array of linearly spaced x values
2. Create a 2-D grid using the x values from step 1 and the y values that were previously defined in task one
3. Assign a variable plot_2D that repeats the 1-D intensity pattern of the double slit to create a 2-D array
4. Plot the figure *Increase the figure size, add a colour map, colour bar and adjust for an aspect ratio of 3:2

In [3]:
```
# task 3
# plotting the 2-D intensity pattern of loght from two slits

x = np.linspace(-1, 1, 10000) # generate an array of 10,000 linearly spaced x values ranging from -1 to 1.
# This grid will be used to plot the 2-D intensity pattern
(X, Y) = np.meshgrid(x,y) # create a 2-D grid using the x values from step 1 and the y values that were previously defined
plot_2D = np.tile(intensity_twoslits, (len(x), 1)) # repeat the 1-D intensity pattern from the variable intensity_twoslits to create a 2-D array
# this replicates the 1-D pattern across the entire grid

plt.figure(figsize=(10,10)) # increase figure size

cmap = plt.cm.gist_heat # select the "gist_heat" colour map for the plot.
# This represents different intensity levels using a range of warm colours.

# plotting the graph, the extent specifies the range of x and y values displayed and the interpolation method is set to 'bilinear' for smooth visualization
c = plt.imshow(plot_2D, cmap, extent =[-6, 6, -6, 6], interpolation ='bilinear')
plt.colorbar(c, orientation='horizontal', label='Light Intensity range') # adding horizontal colour bar to show the range of light intensity
plt.xlabel('y (mm)')
plt.ylabel('x (mm)')
plt.ylim(-6 * 2/3, 6 * 2/3) # creating an aspect ratio of 3:2
plt.title('2-D Intensity Pattern of Light from Two Slits')

plt.show()
```



Task 4

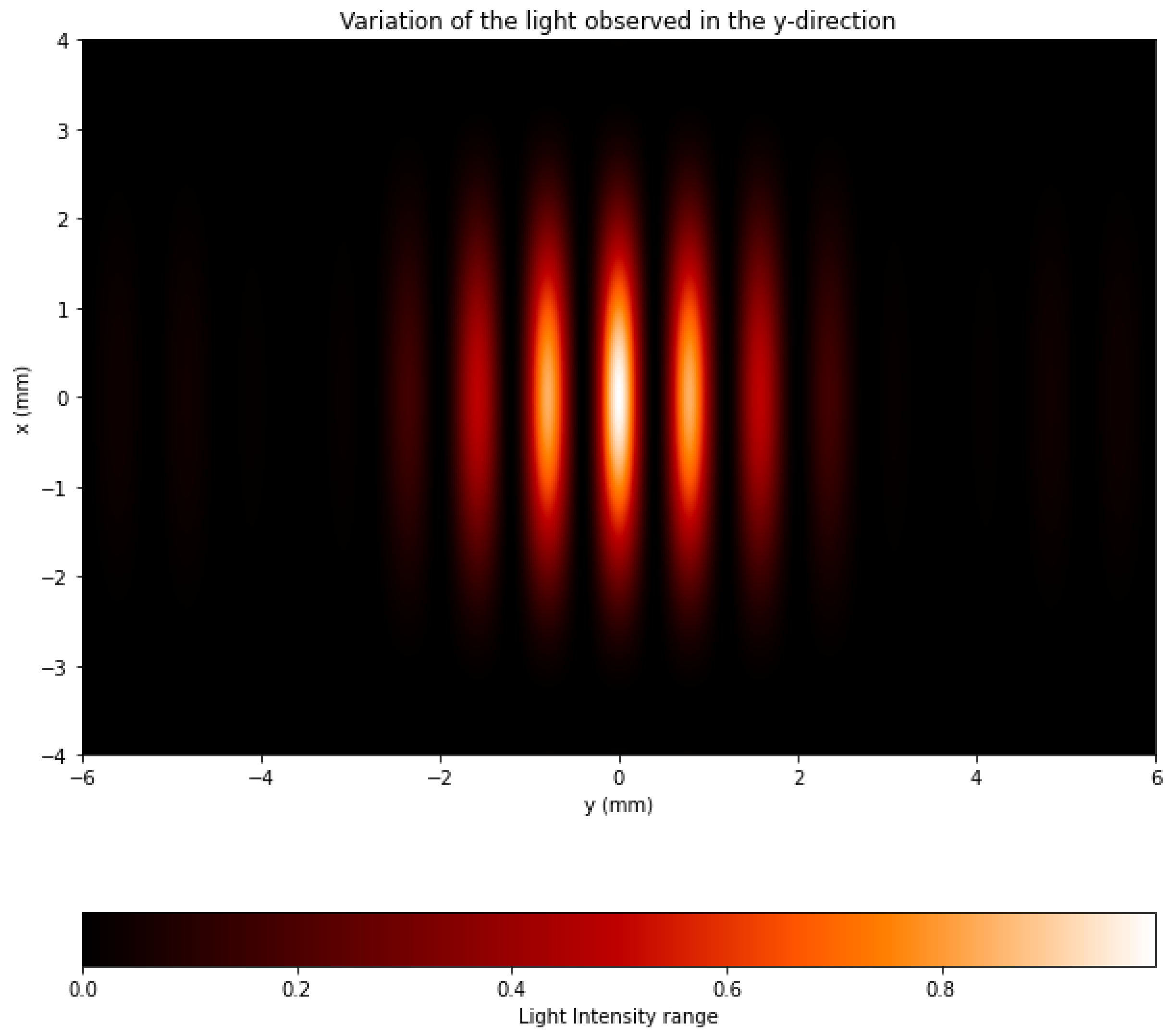
The real variation of the light observed in the y-direction is that it falls off away from the axis. This is because the light undergoes single-slit diffraction in this direction. The purpose of this task is to identify the diffraction part of the formula above to determine an appropriate function for the fall off in intensity in the y-direction, and hence apply it to the 2-D plot. A slit height of b=1 mm is used.

Once again the existing code can be extended and modified. The single slit diffraction effect was determined using a function here, with the y values and slit height b as variables. It returns the single slit intensity pattern of these conditions. A 2-D array was then created using the np.outer method to counter in all the possible permutations of x and y, by taking the outer product of two 1-D arrays.

1. Call a function single_slit_intensity_factor that takes y values and slit height b as variables
2. Assign value to slit height b
3. Create an array for the single slit intensity pattern for the same range of positions
4. Create a 2-D array Z for all outer products of x and y
5. A. Plot the figure *Increase the figure size, add a colour map, colour bar and adjust for an aspect ratio of 3:2

In [4]:

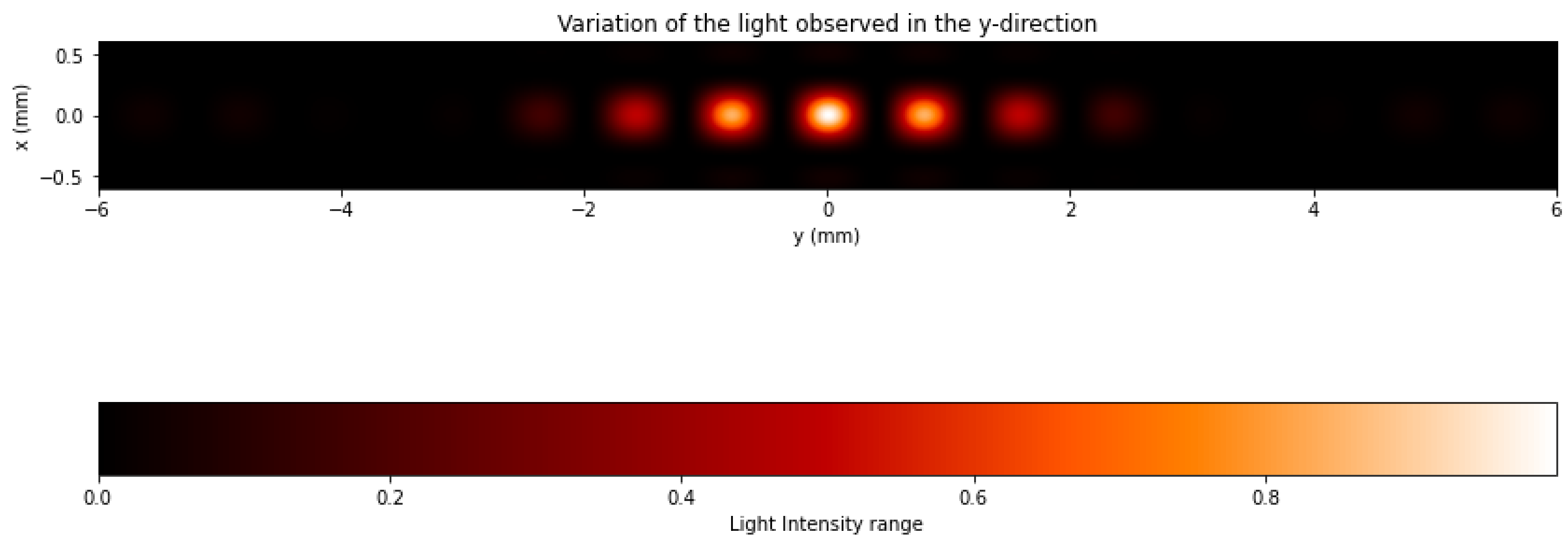
```
# task 4

# calling a function that takes any input y-value and returns the intensity factor for a single slit with a slit height b
def single_slit_intensity_factor(y, b):
     return intensity_oneslit

b = 1  # slit height in millimetres

# create an array for the single slit intensity pattern, for a range of positions [-6,6]
y = single_slit_intensity_factor(np.linspace(-6, 6, 1000), b)

# reuse the double slit intensity pattern defined above
x = intensity_twoslits

# create a 2D array of all outer products of x and y
# x and y are 1D arrays so their product of their permutation is a 2D array
# creating a 'grid' to visualise the interference pattern when single slit diffraction and two slit interference pattern are combined
Z = np.outer(y, x)

# create the plot
plt.figure(figsize=(10,10)) # increase figure size
cmap = plt.cm.gist_heat # selecting same colour map as above

# creating a plot of the Z array , using the same range and interpolation as pervious graph
d = plt.imshow(Z, cmap,  interpolation ='bilinear', extent =[-6, 6, -6, 6])
plt.colorbar(d, orientation='horizontal', label='Light Intensity range') # adding colourbar
plt.xlabel('y (mm)')
plt.ylabel('x (mm)')
plt.ylim(-6 * 2/3, 6 * 2/3) # fixing aspect ratio
plt.title('Variation of the light observed in the y-direction')

plt.show()
```



Changing the slit range/height to make graph look more like desired results

In [5]:

```
# task 4 extended

# create the plot
plt.figure(figsize=(14,14)) # increase figure size
cmap = plt.cm.gist_heat
d = plt.imshow(Z, cmap,  interpolation ='bilinear', extent =[-6, 6, -.6, .6])
plt.colorbar(d, orientation='horizontal', label='Light Intensity range')
plt.xlabel('y (mm)')
plt.ylabel('x (mm)')
#plt.ylim(-6 * 2/3, 6 * 2/3)
plt.title('Variation of the light observed in the y-direction')

plt.show()
```



Here this graph looks more like the desired graph