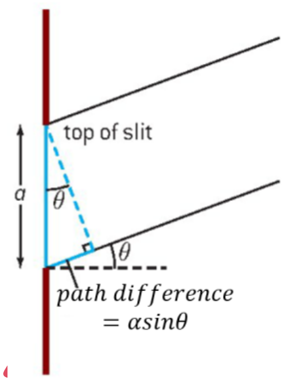
Simulation of

Single and Double-Slit Diffraction Gratings

I used to believe light passing through a hole form the shape of uniform intensity. Not till the study of single/double-slit experiment, did I realize that the light passing through slit(s) generate concentric rings.

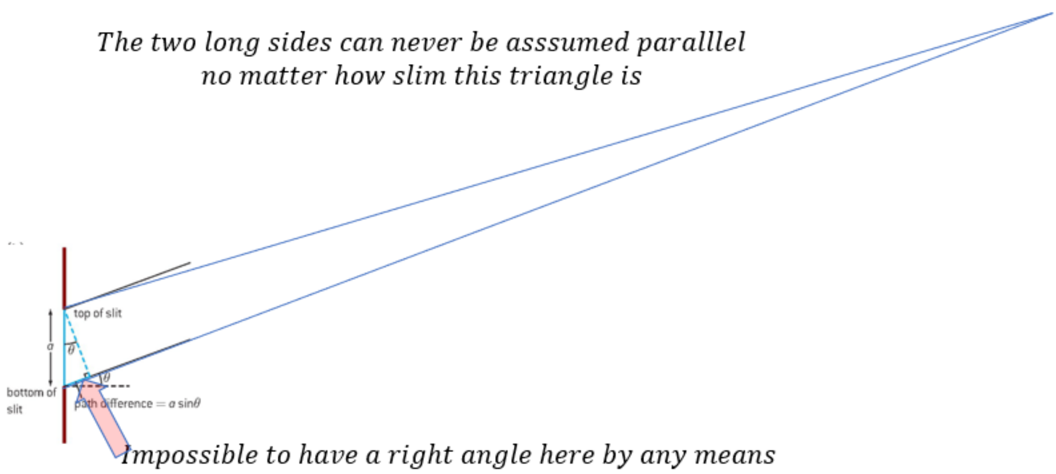
Most textbooks derive the formula in similar ways below.

Troy's Study Note:

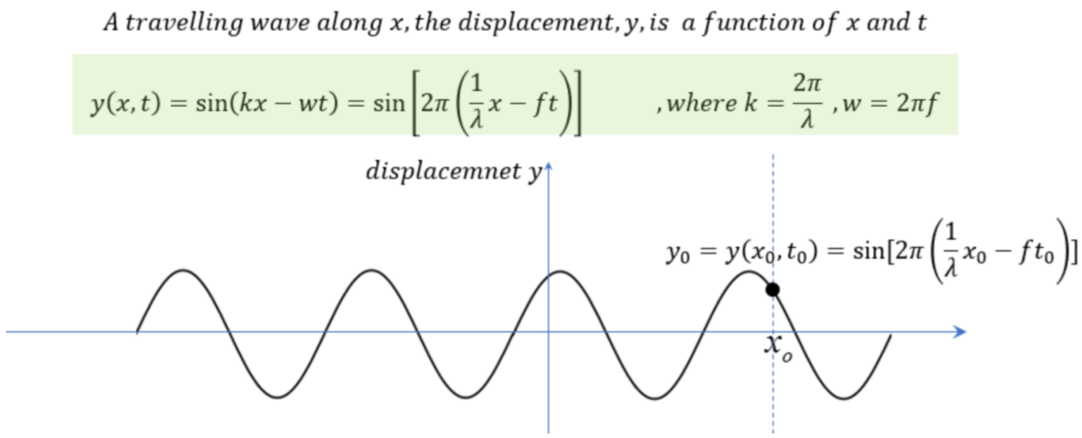


Each point on the wavefront within the slit behaves as a source of waves. These waves interfere when they meet beyond the slit. For the two waves coming from the edges of the slit making an angle θ with the straight through, there is a path difference of $\alpha sin\theta$. Waves from a point halfway along the slit will have a path difference of $\frac{\alpha}{2} sin\theta$ from the waves coming from each of the edges. When this path difference equals half a wavelength, the waves from halfway along the slit will interfere destructively with the waves coming from the bottom edge of the slit. It follows that for each point in the bottom half of the slit there will be a point in the top half of the slit at a distance $\frac{\alpha}{2} sin\theta$ from it. This means that when $\frac{\alpha}{2} sin\theta = \frac{\lambda}{2}$, there is destructive interference between a wave coming from a point in the upper half of the slit and an equivalent wave from the lower half of the slit.

Because it has unreasonable assumption to assume parallel for the two long sides of a triangle, I am not really convinced of the result derived that way. I know it is just to provide intuitive understanding to prevent form complicated mathematics. I decide to convince myself through different methodology.



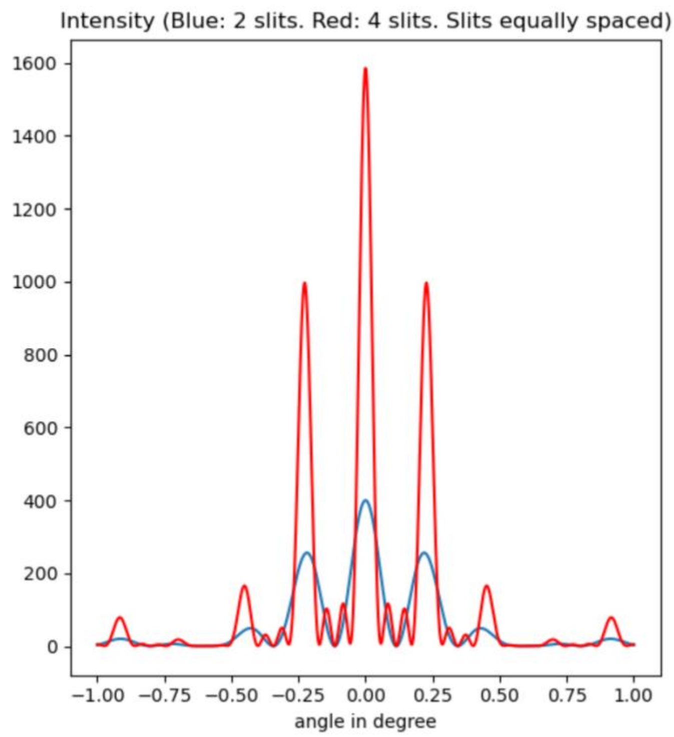
I would like to adopt the fundamental equation describing travelling wave $y = \mathrm{A}\sin(kx - \omega t)\,, k = \frac{2\pi}{\lambda}, \omega = 2\pi f$ to conduct the superposition.



At a fixed position, the oscillating displacement will be

$y = \sum_{i=1}^{\infty} \frac{A}{4\pi L_i^2}\sin(kL_i - \omega t)$, where L_i is the distance from x to source [i], provided all the source have the same amplitude and zero phase offset.

Then I pick the amplitude as the maximum among y values at different time, t, in a period. Then plot intensity (= amplitude square) vs angle goes as below.



I share my source code in python as below.

```
import numpy as np
import matplotlib.pyplot as plt
import matplotlib.image as mpimg    # mpimg for image-reading

print("\n\n\n      Welcome to Troy's program simulating the intensity of multiple slits which have non-negligible width.                              ")

try:
    travellingWaveFormula = mpimg.imread(
        'travelling wave.png')    # read image file in the same directory. YCIS is an np.array now
    plt.imshow(travellingWaveFormula)    # to display image
    plt.axis('off')    # not to show coordinates
    plt.show()
except:
    print(
        "\n\n\n************If you put the 'travelling wave.png' file in the same directory of this code, you can see the fundamental formula this program refers to.********")



# ratioDd=100
D = 1.3    # The radius of circle where light is projected onto (the "D" in IB Physics formula)
d = 150e-6    # the spacing of slits (the "d" in IB Physics formula)
Width = 50e-6    # the width of slit (the "b" in IB formula)
lamda = 600e-9    # wavelength 500nm for blue    and 750nm for red
freq = 3e8 / lamda    # the frequency of light, equal to c/ wavelength
k = 2 * np.pi / lamda    # for the k, =2pi/wavelength, in wave formula: sin(kx-wt)
w = 2 * np.pi * freq    # for the w, =2pi*freq, in wave formula: sin(kx-wt)
print("Parameters about slits: they are spaced %3.2f um with slit width=%3.2f um"%(d/1e-6,Width/1e-6) )

print("\n\nSimulation ongoing . Please wait. Thanks for your patience....")
```

```
    THETA = np.linspace(-1.0, 1.0, 2000)    # the angle away from the vertical bisector of two slits
    theta = (THETA + 90) / 180 * np.pi    # convert THETA to the angle from x-axis amd make it radian
    intensityAtTHETA = theta * 0    # light intensity at each angle

    lightSource1 = np.linspace(-d / 2 - Width / 2, -d / 2 + Width / 2, 10)    # the point sources
distributed over slit#1
    lightSource2 = np.linspace(d / 2 - Width / 2, d / 2 + Width / 2, 10)    # the point sources
distributed over slit#2
    lightSource3 = np.linspace(-d - d / 2 - Width / 2, -d - d / 2 + Width / 2,
                               10)    # the point sources distributed over slit#3
    lightSource4 = np.linspace(d + d / 2 - Width / 2, d + d / 2 + Width / 2,
                               10)    ## the point sources distributed over slit#4
    lightSource = np.append(lightSource1, lightSource2)    # overall point sources over slit#1 and
slit#2

    halfCycle = np.linspace(0, 1 / 2, 60) * 1 / freq    # We will check out half a cycle to decide the
amplitude

    m = 0
    while m < len(theta):
        angle = theta[m]
        amplitude = 0
        xPos = D * np.cos(angle)    # the x coordinate for a position on the circle of radiud D
        yPos = D * np.sin(angle)    # the y coordinate for a position on the circle of radiud D
        n = 0
        while n < len(halfCycle):
            t = halfCycle[n]    # one instant when we are to sum up individual displacements by all
the point sources
            waveSum = 0    # initilization for displacement summation
            p = 0
            while p < len(
                    lightSource):    # We are to sum up (accumulate) all the discplacement from
different point sources, one by one,
                LS = lightSource[p]    # a point source
                disAway = np.sqrt(
                    pow((xPos - LS), 2) + pow(yPos, 2))    # distance away from the source. It is
the "x"    in sin(kx-wt)
```

```
                waveSum = waveSum + np.sin(k * disAway - w * t)    # displacement accumulation using general formula for waves
                p += 1
            if abs(waveSum) > amplitude:    # Find the maximum and make it the amplitude
                amplitude = abs(waveSum)
            n += 1

        intensityAtTHETA[m] = pow(amplitude, 2)    # wave intensity =amplitude*amplitude
        m += 1

    plt.subplot(121)
    plt.plot(THETA, intensityAtTHETA)    # plot intensity over angle
    #plt.plot(THETA, 20*np.log10(intensityAtTHETA))    # plot intensity over angle
    plt.title("Intensity (Blue: 2 slits. Red: 4 slits. Slits equally spaced)")
    plt.xlabel("angle in degree")

    plt.subplot(122)
    plt.plot(intensityAtTHETA*np.cos(theta), intensityAtTHETA*np.sin(theta))    # plot intensity over angle
    plt.title("Intensity (Blue: 2 slits. Red: 4 slits. Slits equally spaced)")
    #plt.xlabel("angle in degree")



    # We are to check out the influence when slits number are increased . We add two more slits below.
    # Only the lightSource is expanded. ALl the rest code are re-used without modification.

    lightSource = np.append(lightSource, lightSource3)
    lightSource = np.append(lightSource, lightSource4)    # overall point sources over slit#1, slit#2, sit#3 and slit#4 (add two slits)

    halfCycle = np.linspace(0, 1 / 2, 60) * 1 / freq    # We will check out half a cycle to decide the amplitude

    m = 0
    while m < len(theta):
        angle = theta[m]
        amplitude = 0
```

```
        xPos = D * np.cos(angle)    # the x coordinate for a position on the circle of radiud D
        yPos = D * np.sin(angle)    # the y coordinate for a position on the circle of radiud D
        n = 0
        while n < len(halfCycle):
            t = halfCycle[n]    # one instant when we are to sum up individual displacements by all
the point sources
            waveSum = 0    # initilization for displacement summation
            p = 0
            while p < len(
                    lightSource):    # We are to sum up (accumulate) all the discplacement from
different point sources, one by one,
                LS = lightSource[p]    # a point source
                disAway = np.sqrt(
                    pow((xPos - LS), 2) + pow(yPos, 2))    # distance away from the source. It is
the "x"    in sin(kx-wt)
                waveSum = waveSum + np.sin(k * disAway - w * t)    # displacement
accumulation using general formula for waves
                p += 1
            if abs(waveSum) > amplitude:    # Find the maximum and make it the amplitude
                amplitude = abs(waveSum)
            n += 1

        intensityAtTHETA[m] = pow(amplitude, 2)    # wave intensity =amplitude*amplitude
        m += 1

    plt.subplot(121)
    plt.plot(THETA, intensityAtTHETA, 'r')    # plot intensity over angle
    #plt.plot(THETA, 20*np.log10(intensityAtTHETA), 'r')    # plot intensity over angle
    plt.subplot(122)
    plt.plot(intensityAtTHETA*np.cos(theta), intensityAtTHETA*np.sin(theta),'r')    # plot intensity
over angle
    #plt.axis([-800, 800, 0, 1600])
    plt.show()
```