Modeling Diffraction patterns using Simpson's approximation of the Fresnel Integral

Peter Spencer-Smith

Introduction:

In this exercise the diffraction intensity pattern resulting from a 1-D single slit and 2-D square aperture is to be modelled.

For part one, a self written Simpson's rule (1) approximation of the 1-dimensional Fresnel integral (2) must be coded.

$$\int_a^b f(x)dx \approx \frac{h}{3}(f(x)_{first} + 4f(x)_{odd} + 2f(x)_{even} + f(x)_{last}) : (1)$$

$$E(x,z) = \frac{kE_0}{2z\pi}\int_{x_1'}^{x_2'} exp\left[\frac{ik}{2z}(x-x')^2\right]dx : (2)$$

The resulting diffraction pattern must then be plotted as a function of screen position, x.

For the second task, the 2-d Fresnel integral (3), which gives the intensity pattern resulting from a square, two dimensional aperture, must be modelled using the Scipy function: *scipy.integrate.simps(yvals, xvals)*.

$$E(x,y,z) = \frac{2z\pi}{kE_0}E(x,z)E(y,z) : (3)$$

This 2-d diffraction pattern must then be plotted as an image.

Simpsons rule takes pairs of points and draws quadratic curves between them, this leads to an increasing accuracy with the number of points. This is superior to the trapezoid rule due to the linear relationship between points and thus many more points are required in order to obtain the same level of accuracy. Simpsons rule requires an even number of intervals and so N+1 points must be taken.

Part One: 1-D Diffraction

The program:

This exercise required an understanding of complex objects in order to model (1) in a clear and straight forward manner.

Equation (2) was converted into polar form and the imaginary coefficient neglected. This meant that the final intensity values could be simply written as (4) without having to use the complex conjugate.

$$I(x,z) = \mathrm{Re}(E(x))^2 + \mathrm{Im}(E(x))^2 : (4)$$

The program structure begins by generating an array of x values which correspond to the screen position.

A set of functions operate on every single value of x and iterate over a range of screen position (x') values using nested for loops. The functions take the real (cosine) and imaginary (sine) terms of (2) and input them into (1) as f(x).
This results in two Simpsons functions, one for the imaginary part and one for the real part, each consisting of the four terms in (1).

These final two terms form the elements in (4) which is plotted as a function of the initial x array.

Results & Discussion:

The intensity pattern generated as a function of screen position -5mm < x < 5mm the aperture, x' = 0.02mm and the screen distance z = 2cm is displayed in figure 1.

Figure 1

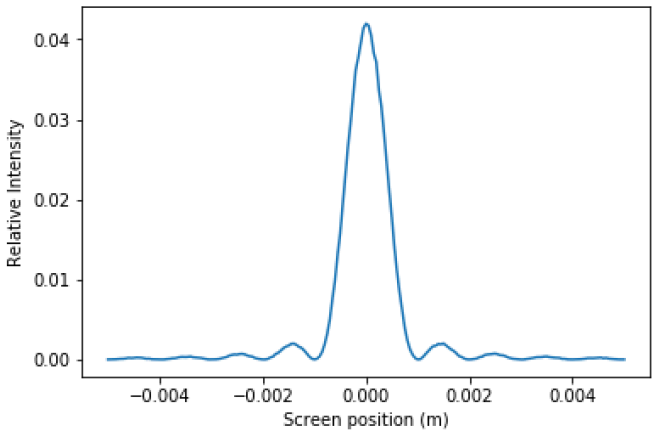


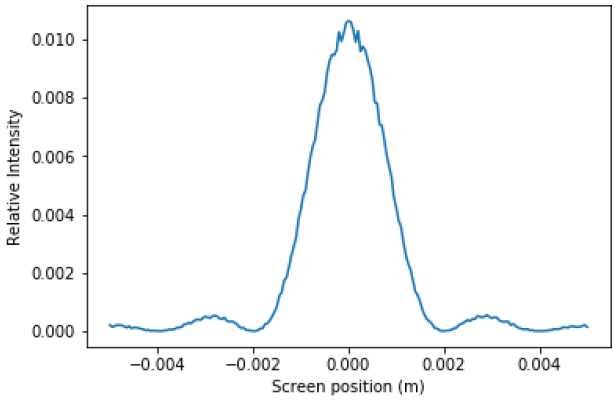
Figure 1 clearly shows the Fraunhofer pattern of a central maximum and n order lower maxima as predicted by the Fresnel integral.

The part one program allows the user to vary z and x'. In doing so we can explore the effects of a changing aperture size and near field effects.

Decreasing x' resulted in wider fringes and decreasing z resulted in narrower fringes, as is conducive with the theoretical predictions and displayed in the figures below.

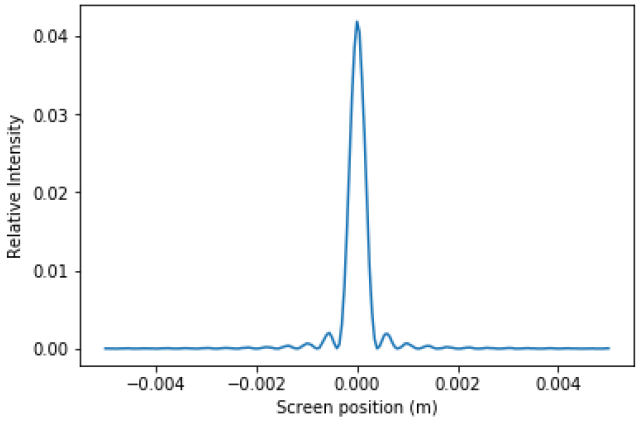
Figure 2 displays the effects of narrowing the slit, where x' = 0.01mm and the remaining parameters are the same as in figure 1.

Figure 2:



Near field effects are displayed in figure 3, where z = 8mm and the number of intervals has been increased to 200 in order to increase the resolution of finer details at this scale.

Figure 3:



Part Two: 2-D Diffraction

Part two proved more challenging, with some unresolved issues, as discussed therein.

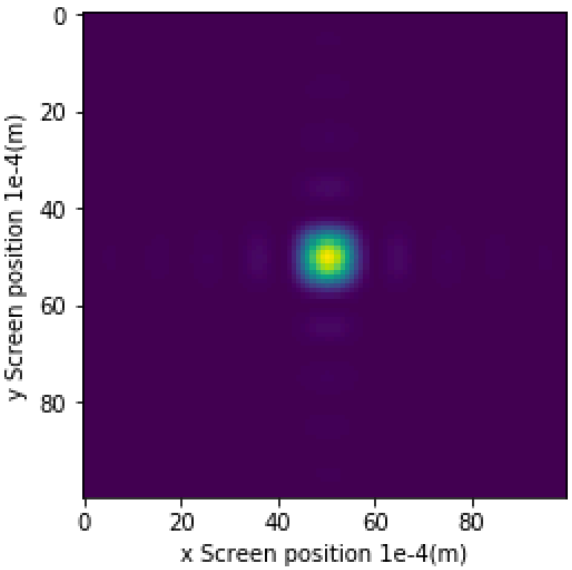
The Program:

In order to generate a two dimensional image, a two dimensional empty matrix was initially generated. This matrix was then filled with values by a function which takes arguments from a set of nested x and y '*for*' loops which are indexed to the matrix. The function operates on each pair of x and y coordinates and generates the intensity values which fill the pixels in the image. It does this by taking every x-y co-ordinate and iterates over the x' and y' aperture dimensions to perform the inbuilt *simps* integral over. The return value in the function models (4), thus giving the intensity on the screen. Care was taken to ensure that the number of intervals was even, in order for Simpson's rule to provide accurate results.

Results & Discussion:

The image presented in figure 4 displays a central maximum with the predicted subsidiary maxima in a cross hair pattern. The image in figure 4 was translated in x and y by 50% in order to centralise the image. x and y translation parameters are present in the program to allow easy change of the image position. This was necessary to overcome an indexing issue that arose from the way the '*for*' loops matched x and y values to the empty matrix. The screen distance, z = 2cm and aperture dimensions, x' = 0.02mm = y' in figure 4 remained unchanged from part one.

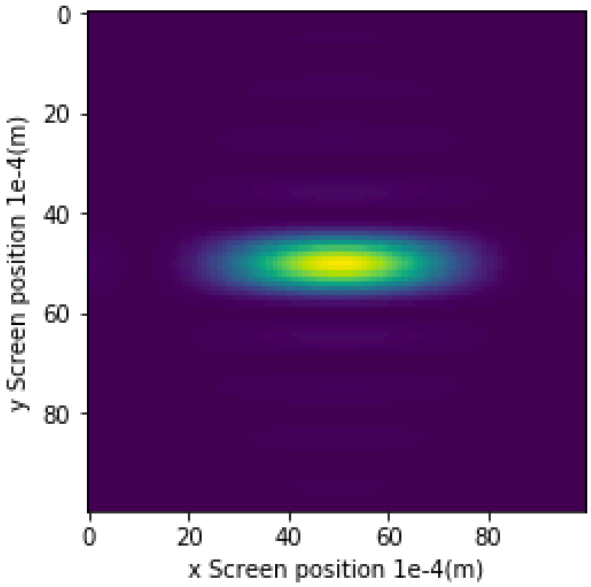
Figure 4:



The part two program allows the user to change the aperture dimensions in order to test the effects of altering them.

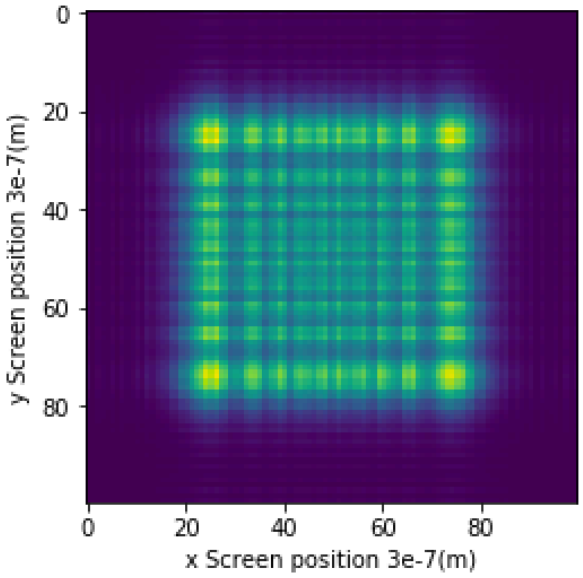
Increasing the aperture size decreases the size of the central maximum and changing the dimensions of the aperture to a rectangular shape causes an inverse distortion of the resulting pattern. Figure 5 displays the results of changing the aperture dimensions to x' = 2e-5 m and y' = 0.5e-5 m.

Figure 5:



Decreasing the screen distance decreased the size of the central maximum, as was analogous to the results in the 1-D experiment. A change in the diffraction pattern was also present as z was gradually decreased.

Figure 6:



In figure 6, z = 0.01mm and the x and y scales were changed to 3e-7 in order to zoom in on the image. These scaling permitters were included to allow straight forward manual change of the image size where necessary.

Improvements:

Issues concerning the screen distance at which the change in diffraction pattern was observed proved difficult to solve. This was most likely due to a number of factors relating to image and scale size rather than a fundamental issue with the way Simpsons rule was implemented. Further investigation is required in order to fully understand this issue.

Efficiency improvements could be implemented in the code in order to speed up the time for the 2-D plots to be generated. Better implementation of the objects being iterated within the for loops could potentially achieve this.

Conclusion:

On the whole the program proved successful. In part one, Simpson's rule was recreated and provided results consistent with theoretical models.

In part two, the central maximum was successfully converted into a two dimensional image and the effects of changing aperture dimension and screen distance were observed . Anomalies in the scales at which these phenomenon were observed proved too difficultly to solve and further investigation in this area is required.