Modelling Fresnel Diffraction Patterns using SciPy Quad Functions.

Angus Brewster

February 2024

Task

Use python to simulate the diffraction pattern, using the Fresnel approximation.

1 History

In modern physics: light, like most other things, is known to exhibit wave particle duality. However, this took many years to understand. One of the first major pieces of evidence for the wave nature of light was the famous Young's double slit experiment in 1801, where 2 slits were illuminated with coherent light which produced a diffraction pattern between the slits when shone onto a screen [2]. This was explained by the 2 light sources interfering which is a property only found in waves. Around 15 years after this a french civil engineer and physicist called Augustin-Jean Fresnel (/freɪˈnɛl/) continued Young's work. Fresnel proposed a mathematical description of the diffraction phenomena. Fresnel's maths was proven correct when Poisson proposed that according to Fresnel's formula there should be a bright spot in the centre of the shadow created by a spherical object. This hypothesis was then proven correct experimentally by Arago, lending credibility to Fresnel's theory. After Fresnel's formula for wavelike light was shown to be true, he also went on to invent the Fresnel lens (a lens made up of a series of small grooves to reduce size while preserving aperture and focusing strength).

Diffraction is a phenomena observed when light interacts with an obstruction. The term Diffraction was first coined by Italian scientist Francesco Maria Grimaldi in 1660 after achieving the first accurate recording of the phenomena [4]. The phenomena of diffraction occuring to a wave that passes through a confined area is due to the Huygens–Fresnel principle, which states that the wavefront of a wave is comprised of a sequence of spherical wavelets which interfere [5]. These wavelets all interfere constructively or destructively depending on their phase difference at different points. If the pattern is shone onto a screen, its pattern will be observed as a series of rings or lines where the wavelets are interfering constructively or destructively known as a diffraction pattern, or spectra.

2 Theory

So what was this theorem that Fresnel proposed? It is now known as Fresnel Diffraction. Fresnel diffraction is observed when either the distance from the source to the obstruction or the distance from the obstruction to the observer is approximately equal in order of magnitude to the obstruction. This is different from previous methods, which required an assumption that the observer was an infinite distance away so that the wavefront could be modelled as planar. In this method the wavefront is modeled a series of spherical wavelets (Huygens' principle) rather than planar. The interference of the wavelets results in different the amplitudes at different locations in the resultant wave. This is observed as a diffraction spectra created by the waves interfering constructively or destructively in different places.

Huygens' principle tells us that the wavefront is made up of many small spherical wavelets, Fresnel's integrals are derived by finding the integral of these wavelets over the aperture [5].

$$C(x) = \int_0^x \cos(\frac{\pi}{2}t^2)dt \tag{1}$$

$$S(x) = \int_0^x \sin(\frac{\pi}{2}t^2)dt \tag{2}$$

The two integrals can be combined to find the electric field at a point on the x' plane

$$E(x,z) = \frac{kE_0}{2\pi z}[C(\frac{x-x'}{\sqrt{2z}}) + i \cdot S(\frac{x-x'}{\sqrt{2z}})] \tag{3}$$

In our code we used Euler's formula to combine the two commonly seen trigonometric Fresnel integrals into one complex exponential form of the integral.

$$E(x,z) = \frac{kE_0}{2\pi z}\int_{x_1'}^{x_2'} e^{\frac{ik}{2z}(x-x')^2}dx' \tag{4}$$

This method can then be expanded to 2d in the $x' - y'$plane:

$$E(x,z) = \frac{kE_0}{2\pi z}\int_{y_1'}^{y_2'}\int_{x_1'}^{x_2'} e^{\frac{ik}{2z}(x-x')^2(y-y')^2}dx'dy' \tag{5}$$

3 Part A

Equation 4 can be solved in python by using SciPy's module quad. As this is a finite integral, quad calls the Quadpack routine from the Fortran library. Quadpack uses the Clenshaw-Curtis method, first it changes the limits to 1, then the function is approximated by a finite Chebyshev series. The coefficients of this series expansion are then calculated and the expanded function is summed with the first and last terms halved, resulting in a numerically solved integral [3]. To find the diffraction pattern equation (4) was integrated for a series of x values and plotted against the relative intensity, which is proportional to the absolute of the field strength squared.

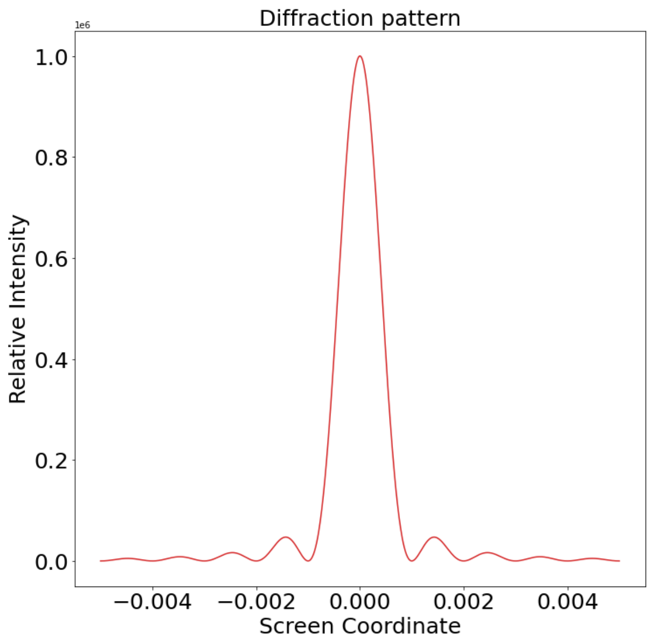

Figure 1: Intensity vs x coordinate on screen for aperture width of 2E-5, distance of 2E-2 (1000x1000 grid).

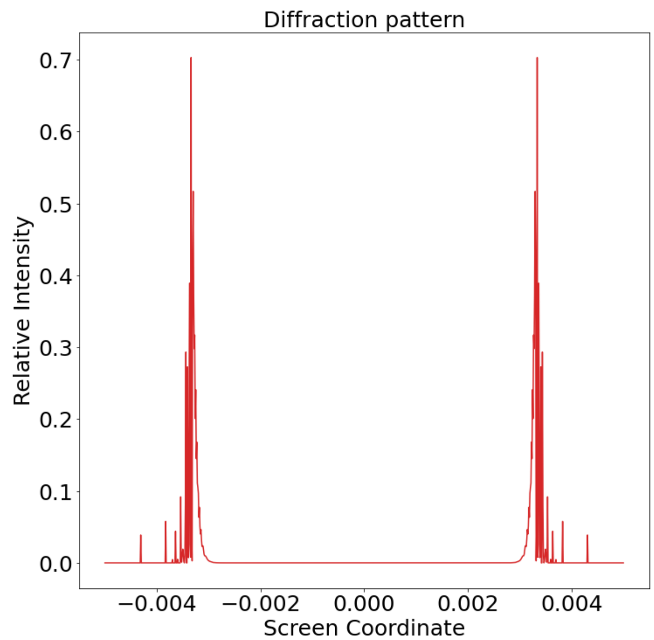

Figure 2: Plot of Errors in Intensity vs x coordinate on screen for aperture width of 2E-5, distance of 2E-2 (1000x1000 grid).

From this output, the diffraction pattern can be seen clearly. To investigate the accuracy of this plot, the errors were also plotted, from this we can see after about $|x| > 0.003$ there is a massive increase in errors this is where the calculation has a much lower output, so perhaps the integration method is less accurate as x tends to 0. Quad also has the ability to adjust its accuracy settings through parameters, but for no value in our testing did it have a notable impact on our outcome.

Next we investigated the impact of varying Aperture width and screen distance on the 1 dimension plot:

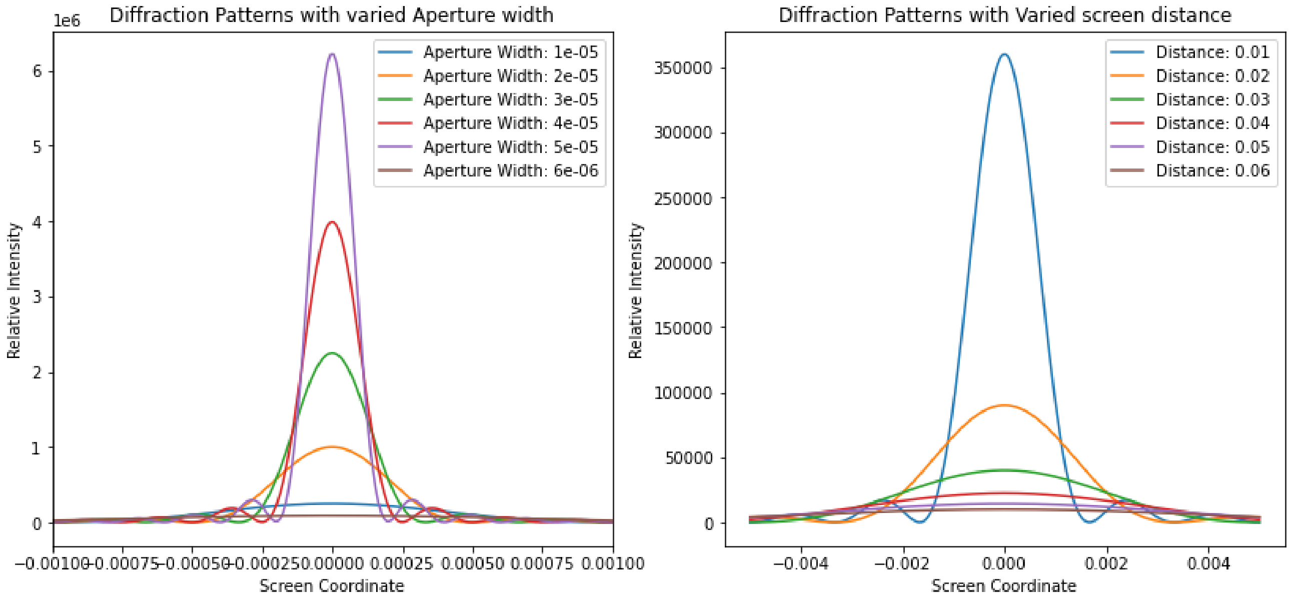

Figure 3: Plot of diffraction patterns: screen coordinate vs relative intensity, with varying aperture width and varying screen difference (100x100 grid).

From these plots, you can see that with both lower aperture width and screen distance, the peak of the pattern has a higher relative intensity with a relatively smaller width. The reduction in the amplitude of the peak for increasing distance could be explained by the inverse square law, where the intensity of light reduces at a rate proportional to the inverse square of its distance to the source. This is correct because in equation (4) $\frac{1}{z}$ is a constant and what is being plotted is the intensity of the wave, which is the absolute squared of the calculated value. Similarly the aperture width has the result of increasing the integration range, which also increases the output based on a constant, resulting in a similar behaviour.

4 Part B

Next equation (5) was solved using the dblquad function which uses the exact same method as the quad function but calculates it twice [1]. Once for each integration variable, in this case x' and y'. The output of this is then plotted in a colour map:

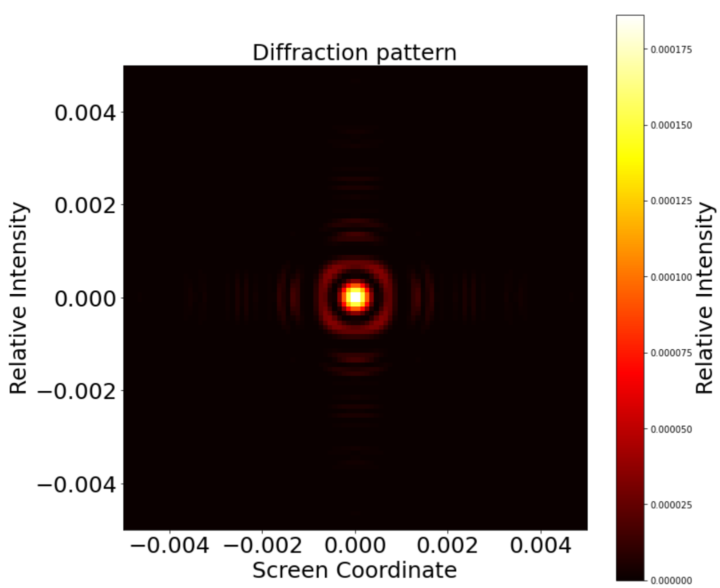

Figure 4: Heat map of intensity for a Fresnel diffraction in 2d (100x100 grid).

This from this plot the shape of the aperture can be seen as a square.

4.1 Varying NumPoints

Next the impact of changing the number of points (”NumPoints”) was investigated. Bellow is a variety of plots with resolutions:

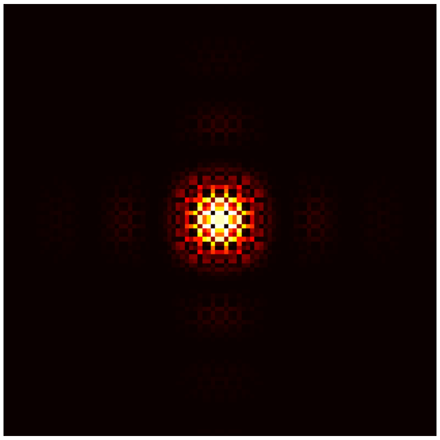

Figure 5: 2d Square diffraction pattern (100x100 grid).

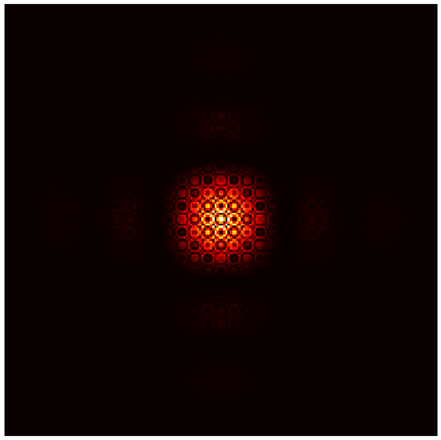

Figure 6: 2d Square diffraction pattern (200x200 grid).

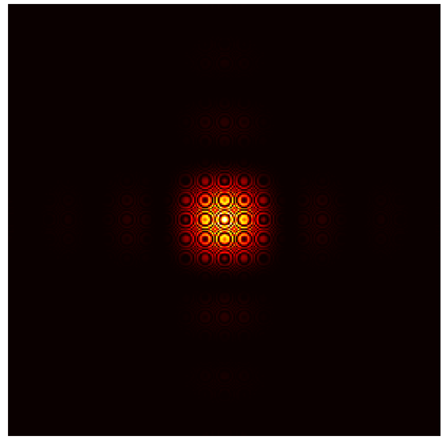

Figure 7: 2d Square diffraction pattern (300x300 grid).

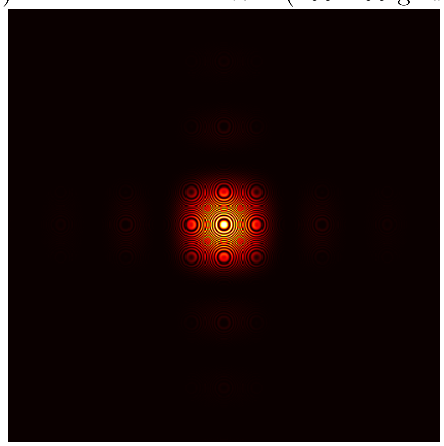

Figure 8: 2d Square diffraction pattern (500x500 grid).

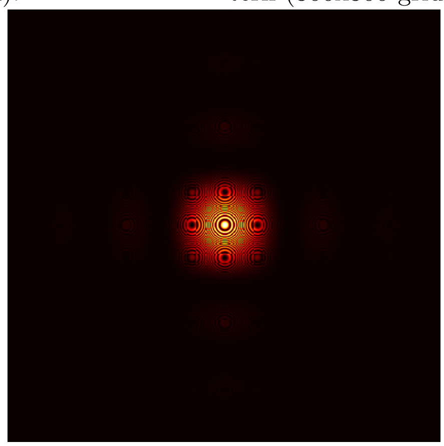

Figure 9: 2d Square diffraction pattern (1000x1000 grid).

From this you can see that as the resolution improves, the image becomes clearer. There are also some circles that appear, this seems to be as a result of the integration method as they change randomly as the resolution changes.

4.2 Varying Distance

Next the diffraction was calculated for different screen distance values:

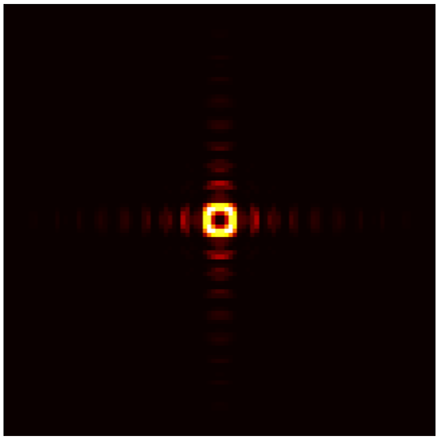

Figure 10: 2d Square diffraction pattern with screen distance of 1cm (100x100 grid).

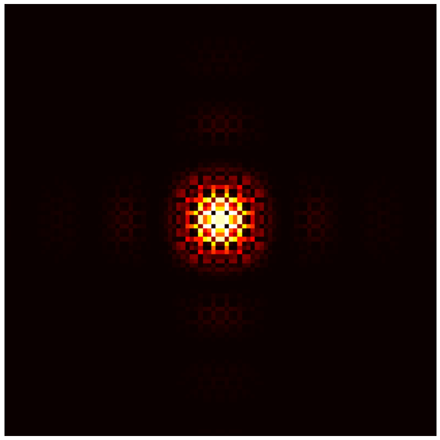

Figure 11: 2d Square diffraction pattern with screen distance of 2cm (100x100 grid).

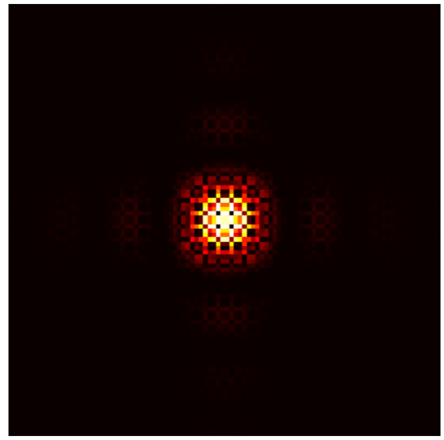

Figure 12: 2d Square diffraction pattern with screen distance of 3cm (100x100 grid).

similarly to the 1d diffraction, as the distance becomes larger, the peak becomes less bright and increases in width.

4.3 Varying ApertureWidth

The aperture width was then varied for more calculations:

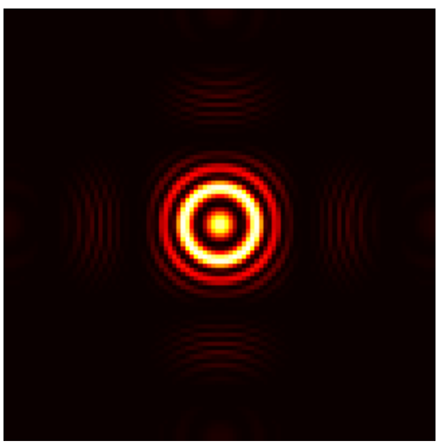

Figure 13: 2d Square diffraction pattern with aperture width of 1E-5m (100x100 grid).

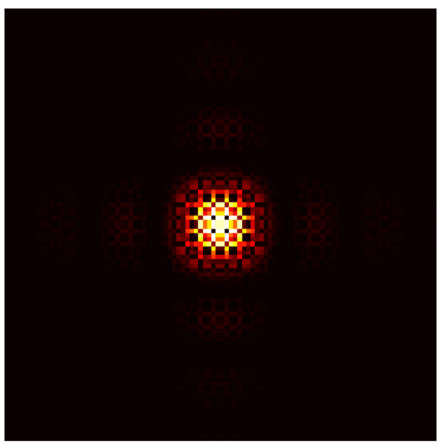

Figure 14: 2d Square diffraction pattern with aperture width of 2E-5m (100x100 grid).

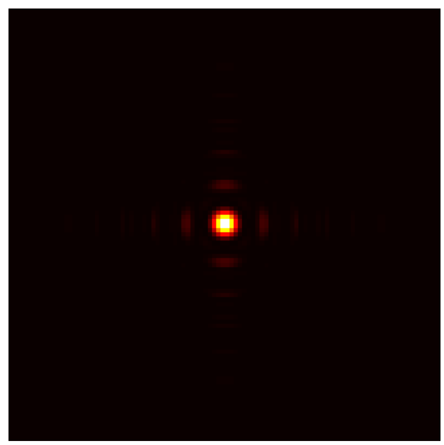

Figure 15: 2d Square diffraction pattern with aperture width of 3E-5m (100x100 grid).

Increasing the aperture width has the inverse effect as the screen distance. Ass it was increased the peak would become more narrow and brighter.

5 Part C

5.1 Using a Rectangular Aperture

This plot could also be transformed to be used for a rectangular integral, by having the x_prime$_{1\&2}$ and yprime$_{-1\&2}$ variables adjust differently according to the aperture width by parametrising a rectangle, this resulted in yprime being a function of x. The use of the lambda argument was required in the definition of the limits in dblquad to show it is a function of the other limit, and not a constant.

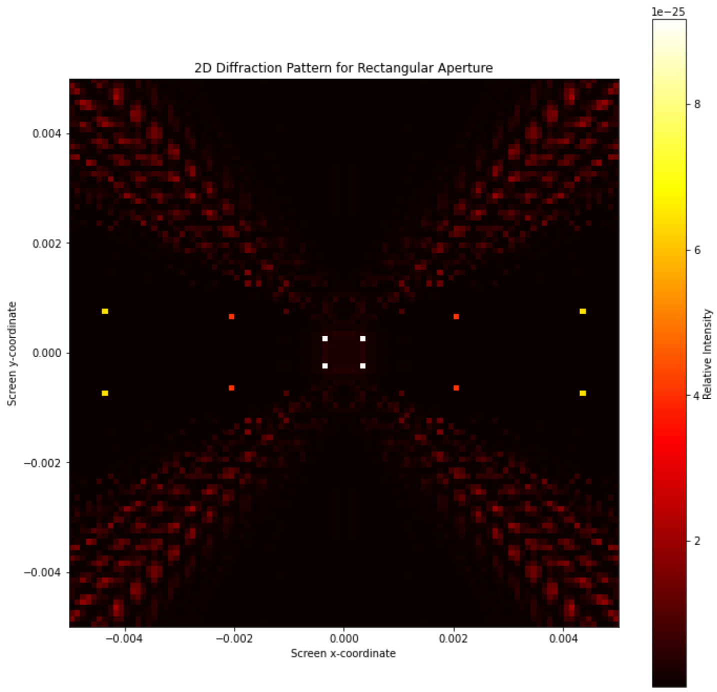

Figure 16: Diffraction pattern for a rectangular aperture (100x100 grid).

5.2 Using a Circular Aperture

The plot could also be done using a circular aperture by using a parameterization of a circle. Again the lambda argument was used as y is a function of x. unfortunately this plot took a very long time to produce so was only printed with low resolution:

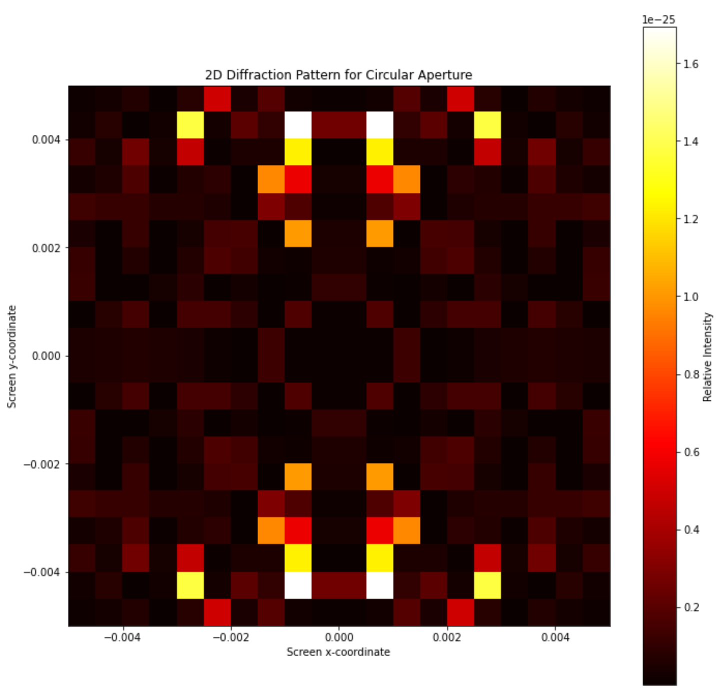

Figure 17: Diffraction pattern for a circular aperture (20x20 grid).

5.3 Changing the integration order

Next we investigated the impact of changing the integration order. The same method as Part B was used however the integral was done with y as a function of x instead of as x as a function of y and the integration order was flipped. For our initial integral this makes no difference as a square aperture's y is not a function of x, resulting in an identical output. The same thing occurred with the circular aperture as it is symmetrical for y and x. The rectangular aperture had the result that the output image appears to be reflected in the x axis creating a sort of cross:

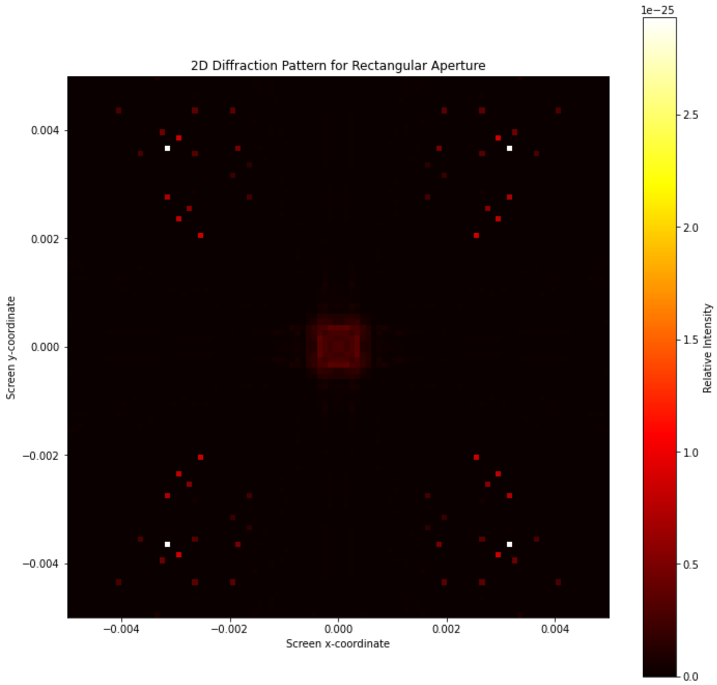

Figure 18: Diffraction pattern for a rectangular aperture, with integration order flipped (100x100 grid).

6 Potential improvements

The code could be improved in a number of ways.

- Further modularisation would be more efficient and avoid repeats, such as having a module to plot, to calculate the intergrand etc.
- less reliance on globals would stream line the code, achievable through the use of parameters.
- A GUI would provide an easier to understand user interface.
- The plotting of the diffraction patterns would have benefited from rendering faster.

The later issue could have been solved by using a common technique, GPU encoding. GPU, rather than the default CPU encoding, is much more efficient at calculation visuals. There is currently no way to do this using the Matplotlib library, thus our code would have to be majorly edited. One such way of doing this in python is with VisPy, which takes advantage of the OpenGL backend to render with a GPU. VisPy even has a built in function called plot which is used to plot graphs, similar to Matplotlib. VisPy was not used for this task as it does not come installed with a standard build of python meaning the examiner would have to install it. To make the user experience better, a gui could be have been used rather than input through the console. This could be achieved in python by using modules such as Tkinter which, opens a new window to display a GUI that can be interacted with using the mouse.

References

[1] Scipy library main repository. https://github.com/scipy/scipy (accessed at 12 february 10:51am).

[2] C. Aime, É. Aristidi, and Y. Rabbia. The fresnel diffraction: A story of light and darkness. *European Astronomical Society Publications Series*, 59:37–58, 2013.

[3] H. O'Hara and Francis J. Smith. Error Estimation in the Clenshaw–Curtis Quadrature Formula. *The Computer Journal*, 11(2):213–219, 08 1968.

[4] Jean-Pierre Revol. Diffractive physics at the cern large hadron collider. *Nuclear Physics A*, 862-863:212–222, 2011. The Sixth International Conference on Physics and Astrophysics of Quark Gluon Plasma (ICPAQGP-2010).

[5] F.A. Volpe, P.-D. Létourneau, and A. Zhao. Huygens–fresnel wavefront tracing. *Computer Physics Communications*, 212:123–131, 2017.