



University College Dublin

An Coláiste Ollscoile, Baile Átha Cliath

PHYC30170 Physics Astro and Space Lab 1

Diffraction Pattern due to a Rectangular Aperture

20 September 2025

by Joana C.C. Adao (Student No. 23311051)

Contents

Abstract

1	Introduction	1
2	Theory	1
2.1	The Wave Nature of Light	1
2.1.1	Spherical Waves	1
2.2	Huygens-Fresnel Principle	2
2.3	Diffraction	3
2.3.1	Fraunhofer vs. Fresnel Diffraction	3
2.3.2	Diffraction for a Single Slit	4
2.3.3	Diffraction for Two Slits	5
2.3.4	Diffraction for a Circular Aperture	5
3	Experimental Setup & Procedure	6
3.1	Rectangular Aperture — Single Slit	6
3.2	Rectangular Aperture — Double Slit	6
3.3	Circular Aperture	7
3.4	Plots and Graphs	7
4	Results	8
4.1	Rectangular Aperture — Single Slit	8
4.1.1	Vertical Aperture	8
4.1.2	Vertical Aperture — Varying Distance to Screen	9
4.1.3	Horizontal Aperture	10
4.2	Rectangular Aperture — Double Slit	11
4.3	Circular Aperture	12
5	Analysis & Discussion	13
5.1	Single Slit Diffraction	13
5.2	Double Slit Diffraction	14
5.3	Circular Aperture Diffraction	14
6	Conclusion	15
References		16
List of Figures		17
Appendix		18

Abstract

The diffraction patterns formed through rectangular and circular apertures with the Huygens-Fresnel principle are considered and discussed through computational simulations of the intensity profiles and two-dimensional pattern simulations for a single slit, double slit, and circular aperture. The results obtained for the Fresnel (near-field) predictions were then compared to the Fraunhofer (far-field) approximations graphically. The computational approach reproduced the main qualitative features of diffraction that include a central maxima, destructive and constructive interference fringes, and an enclosing envelope that is wavelength-dependant. The experiment reveals that the far-field approximations by Fraunhofer do not account the Fresnel effects observed in the simulations conducted. The study goes on to show the validity of the Fraunhofer approximations in the approaching far-field regime while also verifying the framework provided by the Huygens-Fresnel principle in diffraction pattern formation.

1 Introduction

The wave nature of light allows for the diffraction or "spread out" of the electromagnetic waves when they encounter a barrier or obstacle, such as an aperture, of similar scale to the wavelength. The Fresnel (near-field) and Fraunhofer (far-field) diffraction regimes are the most common ways of describing the patterns observed. The Fraunhofer mathematical expressions for diffraction are typically simpler, thus often used to explain theory, yielding an intensity profile of the sinc^2 form for a single slit or the \cos^2 -modulated form for the double slit. However, these approximations do not hold at finite close distances to the screen. [1–3]

In this report, the pattern and behaviour of diffraction is studied computationally with consideration for the Huygens-Fresnel principle. Simulations of the rectangular apertures for the single and double slits, as well as of a circular aperture, were performed and compared with the corresponding Fraunhofer approximations. This approach allowed for the verification of the theoretical context and exploration of the Fresnel effects that arise with finite distances and wavelengths.

2 Theory

2.1 The Wave Nature of Light

Light is an electromagnetic wave. It is a transverse wave in such that it displaces the medium perpendicular to the direction of propagation. Although the wave moves through the medium, the medium itself is not displaced by the wave, and such the atoms themselves remain near their equilibrium positions. This is one of the significant properties that distinguishes waves from a stream of particles. [2]

2.1.1 Spherical Waves

An idealised point source of light emits radiation radially outwards and evenly in all directions, forming a spherical wavefront as shown in Figure (1). The source is then described as isotropic, producing wavefronts that expand as concentric spheres increasing in diameter with distance from the source. [2]

For a spherically symmetrical description of this wavefront the descriptive function must depend only on the radial component r such that [1, 2]:

$$\psi(\vec{r}) = \psi(r, \theta, \phi) = \psi(r) \quad (1)$$

and for a harmonic spherical wave progressing radially outwards from the origin, the wave function is given by [2]:

$$\psi(r, t) = A \frac{\cos(kr - vt)}{r} \quad (2)$$

with a constant speed v and A as the source strength, converging toward the origin. At any given value for time, this function corresponds to a set of concentric spheres extending through all space. [2]

The electric field equation at position r for a spherical wave of monochromatic light originating from a point source at position r_1 is given by [4]:

$$E(r, t) = \sum_{i=1}^N A \frac{\cos(k|r - r_1| - \omega t)}{|r - r_1|} \quad (3)$$

where $|r - r_1|$ is the distance from the point source to the observation point, $k = \frac{2\pi}{\lambda}$ is the wave number, λ is the wavelength of light, $\omega = 2\pi f$ is the angular frequency and f is the frequency of light, with N wavelets superimposed. [4]

This function for the wave can also be described in its complex form [2]:

$$E(r, t) = \left(\frac{A}{|r - r_1|} \right) e^{i(k|r - r_1| \pm \omega t)} \quad (4)$$

The intensity observed corresponds to the time-averaged square of the electric field such that [4]:

$$I \propto |E|^2 \quad \Rightarrow \quad \overline{E^2} = \frac{1}{T} \int_0^T E^2 dt \quad (5)$$

Equation (2) gives the general spherical wave solution, Equation (3) applying the same form to an electric field of light. Both show a dependency of amplitude A on $\frac{1}{r}$, a diminishing amplitude with increasing distance.

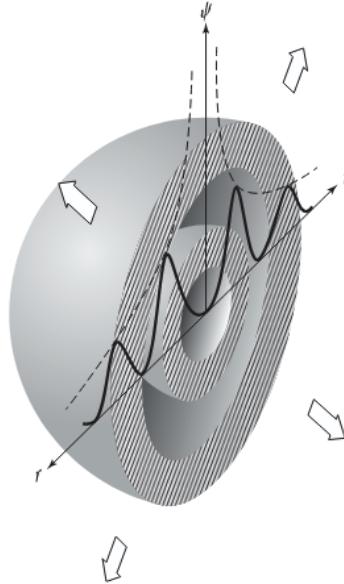


Figure 1: Visual representation of a spherical wavefront from a point source; graph illustrates the amplitude decreasing with distance as the wave propagates radially outward. [2]

2.2 Huygens-Fresnel Principle

Huygens Principle, or more commonly known as the "Huygens-Fresnel Principle", states that each unobstructed point on a wavefront can be thought of as an individual source of secondary spherical wavelets, each oscillating at the same frequency as the original wave, as seen in Figure (2). [1, 2, 5, 6]

The resulting optical field at any later point is then obtained by summing all the wavelets and taking into account their relative phases and amplitudes, building upon the superposition principle idea. [1, 2]

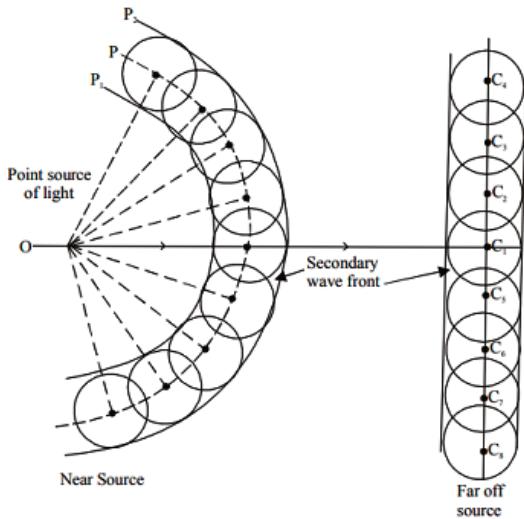


Figure 2: Diagram of Huygens' Principle for a near source (left) and far off source (right), showcasing the secondary wavelets at the secondary wavefront. [7]

Applying this idea qualitatively it can be understood that for large wavelengths relative to an aperture, the waves spread widely into the region past the obstruction through diffraction. Therefore, as the aperture size decreases, the diffracted waves take on an increasingly circular form and only then will the wavelets interfere constructively. [1, 2]

2.3 Diffraction

Diffraction is the phenomenon of a wave, such as light, sound, or matter, bends or spreads out as it encounters an obstacle. When part of a wavefront is blocked or altered in amplitude or phase, the waves that pass beyond that obstacle overlap and interfere with each other creating what is known as a **diffraction pattern**. [1, 2]

2.3.1 Fraunhofer vs. Fresnel Diffraction

Expanding on the concept of diffraction, when a light is shined through a single small aperture in a screen, the light will make an image of the hole on another screen placed closely behind the first one with very faint surrounding fringes. As the distance between the distance between both screens increases, the fringes become more visible, increasing in strength and detail, while the image of the hole is still prominent. This effect is known as **Fresnel** or **near-field** diffraction. [2, 5, 8]

If the second screen continues to be moved farther away, the fringe pattern observed will change gradually. At a very large distance, the pattern spreads out so much that it bears very little resemblance to the original aperture. This effect is known as **Fraunhofer** or **far-field** diffraction. [2, 8]

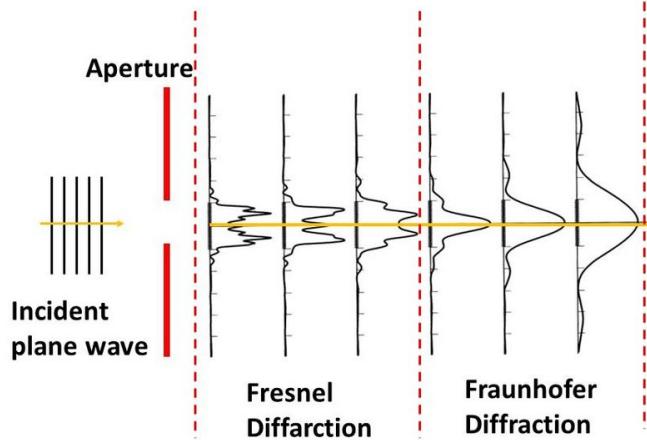


Figure 3: Intensity profiles in the near-field diffraction (middle) compared with the far-field diffraction (right) intensity profiles. [9]

Both Fresnel and Fraunhofer diffraction can be understood as direct consequences of the Huygens-Fresnel Principle (§2.2). In the near-field, the curvature of the spherical wavelets is considered, giving rise to the well-defined visible fringe patterns, while in the far-field the wavelets can be approximated to plane waves, leading to simpler and less defined diffraction patterns, see Figure (3). [2]

2.3.2 Diffraction for a Single Slit

When coherent monochromatic light passes through a narrow slit it spreads out to form fringes on a screen. The pattern consists of a bright central maximum that's significantly wider and more intense than the peripheral fringes. A series of alternating dark fringes (minima) and bright fringes (maxima) appear to either side. The peripheral maxima are only a small fraction of the intensity of the central maximum that decreases rapidly with increasing distance to the centre, see Figure (4). [1–3]

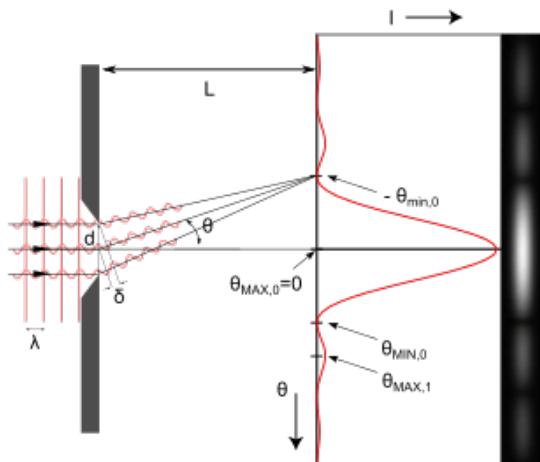


Figure 4: Single slit diffraction diagram; infinitely many points (three) shown along length d project phase contributions from the wavefront, producing a continuously varying intensity θ on the registering plate. [10]

2.3.3 Diffraction for Two Slits

When coherent monochromatic light passes through two closely spaced narrow slits the singular circular wavefront is split into two, each slit acting as a secondary source for the wavefronts that spread out and overlap. The new secondary wavefronts maintain coherency and a constant phase relationship as they are derived from the same initial wave. The superposition of these wavefronts produces a pattern of alternating dark and bright fringes when projected onto a screen. The central maximum remains the brightest and most intense, with the fringes decreasing in intensity with increasing distance from the centre (as with the single slit), see Figure (5). This diffraction pattern was first demonstrated in Thomas Young's 1801 experiment. [2, 3, 11]

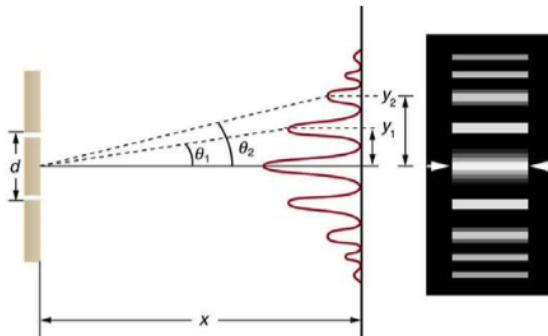


Figure 5: Double slit diffraction diagram; intensity falls off with angle to form an interference pattern consisting of a series of bright and dark fringes. [3]

2.3.4 Diffraction for a Circular Aperture

When coherent monochromatic light passes through a small circular aperture the pattern that appears is of a bright central spot maxima surrounded by fuzzy, airy concentric rings decreasing in intensity with increasing distance from the centre. The central disc is the most intense and the intensity of the surrounding rings is dependant on the size of the aperture. Rings become more regular and defined in the far-field (Fraunhofer) regime, notably more complex and less defined when in the near-field (Fresnel) regime, see Figure (6). Deviations from the usual pattern arise if coherence is not perfect, if the aperture size is comparable to the wavelength, or if the illumination is non-uniform. [3, 12–14]

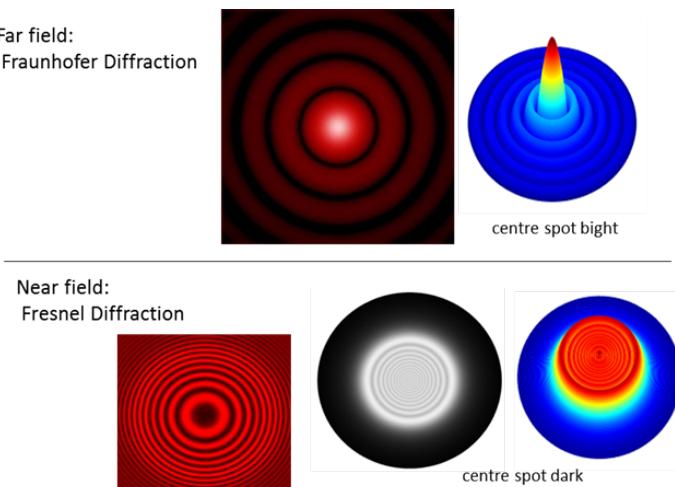


Figure 6: Circular aperture diffraction pattern; Fraunhofer (far-field) diffraction (top), Fresnel (near-field) diffraction (bottom) where a central black spot can be observed. [15]

3 Experimental Setup & Procedure

For this experiment, the standard numpy and matplotlib.pyplot packages were used, in addition to matplotlib.colors for aesthetic reasons, find_peaks imported from scipy.signal to indentify and plot peaks and troughs in resulting graphs, and the j1 function imported from scipy.special for the use of in theoretical plots for the circular aperture (see Appendix, lines 2–6).

The parameters for the experiment are set up, particularly the width (w) and height (h) of the slit, distance from the slit to the screen (L), the wavelength of light (lam) and the corresponding wave number/vector (k), calculated with $\frac{2\pi}{\text{lam}}$ (see Appendix, lines 21–25).

To define an aperture size a proper resolution (nx, ny) must be chosen, decided from the ratio between the width and height of the slit being used. For a 1mm x 3mm slit the ratio is 1:3, therefore the resolution would be nx and 3nx. The screen size is then defined spanning all width and height from $-\frac{w}{2}$ to $+\frac{w}{2}$ and $-\frac{h}{2}$ to $+\frac{h}{2}$ respectively, divided into the points defined with nx and ny. These points are overlaid and equally spaced to create an even array of data (see Appendix, lines 28–35).

3.1 Rectangular Aperture — Single Slit

For a rectangular aperture, a mask over which the code acts over is defined with the condition $|x| \leq \frac{w}{2}$ and $|y| \leq \frac{h}{2}$, and these values are flattened in the x- and y-direction and stacked to create a single symmetrical array for computing (see Appendix, lines 37–42, 218–232).

To calculate the intensity profile curve of the diffraction pattern Equation (4), omitting the time component as a monochromatic wave source does not change with time. The total electric field is summed and used to find intensity using the relationship in Equation (5), added into an empty array and normalised (see Appendix, lines 46–60, 236–250).

To plot the theoretical Fraunhofer plot lines for comparative analysis a new formula is introduced, with a as width of the slit (see Appendix, lines 62–72, 252–262) [3]:

$$I = I_0 \left(\frac{\sin \beta}{\beta} \right)^2 ; \quad \beta = \frac{\pi a \sin \theta}{\lambda} \quad (6)$$

Additionally, the theoretical Fresnel plot lines can also be added with the following equation, but through comparative analysis it is deduced that the code matches theory (see Appendix, lines 74–91, 264–281) [16]:

$$E(x, y, z) = \frac{e^{ikz}}{i\lambda z} \iint_{-\infty}^{+\infty} E(x', y', 0) e^{\frac{ik}{2z}[(x-x')^2 + (y-y')^2]} dx' dy' \quad (7)$$

To simulate the 2D diffraction pattern that would be observed on the screen the same Equation (4) and relationship Equation (5) is taken, redefining over a two-dimensional screen with n, m resolution points that are stacked to create a single two-dimensional grid of the simulated area over which the electrical field and intensity are calculated and summed over (see Appendix, lines 93–112, 283–302).

3.2 Rectangular Aperture — Double Slit

For the double slit computational experiment a similar setup to the single slit is followed, adding a separation variable (sep) and adjusting the values of the width (w), the ratio between

a 0.2mm x 3mm slit with a 0.6mm separation (from left edge of left slit to right edge of right slit) is then a resolution of 1:1.9 and therefore nx and $1.9nx$. The mask for the double slits is taken for each separate slit and then considered with the bitwise "OR" python component, i.e. one or both masks must agree on a position for it to be added to the array (see Appendix, lines 401–425).

The intensity profile curve and two-dimensional code setup remains the same as the single slit setup (see Appendix, lines 429–443, 477–496).

For theoretical Fraunhofer plot lines for comparative analysis a term of $\cos^2 \alpha$ is added to Equation (6) to become, with b as the slit separation (see Appendix, lines 445–456) [17]:

$$I = I_0 \left(\frac{\sin \beta}{\beta} \right)^2 \cos^2 \alpha \quad ; \quad \beta = \frac{\pi a \sin \theta}{\lambda} \quad , \quad \alpha = \frac{\pi b \sin \theta}{\lambda} \quad (8)$$

The same setup for the theoretical Fresnel approximation for a comparative analysis is used as for the single slit (see Appendix, lines 458–475).

3.3 Circular Aperture

For the circular aperture computational experiment a similar setup to the single and double slits is followed, adjusting the values of the width (w) and height (h) by adding the radius (R) such that $w = h = 2R$, the ratio between therefore becomes 1:1 so nx and ny are the same resolution value. The mask for the circular aperture is taken for the circle inequality ($x^2 + y^2 \leq R^2$) and proceeded with in the same fashion as the single slit (see Appendix, lines 595–617).

The intensity profile curve and two-dimensional code setup remains the same as the single slit setup (see Appendix, lines 621–635, 672–691).

To plot the theoretical Fraunhofer lines a different expression is used, using the Bessel function J_1 to calculate the airy disc pattern (see Appendix, lines 637–650) [18]:

$$I = I_0 \left(\frac{2J_1(kR\theta)}{kR\theta} \right)^2 \quad (9)$$

The same setup for the theoretical Fresnel approximation for a comparative analysis is used as for the single slit (see Appendix, lines 652–670).

3.4 Plots and Graphs

All graphs were plotted in a similar way: an intensity profile graph with theoretical Fraunhofer and Fresnel lines (top left), a two-dimensional simulation of the pattern produced (bottom left), marked and labelled peaks and throughs on the original intensity profile plot (top right), and a semi-log plot of the intensity decay of the diffraction patterns with the original intensity profile for comparison (bottom right).

4 Results

4.1 Rectangular Aperture — Single Slit

4.1.1 Vertical Aperture

With a set screen distance of 200mm from a rectangular aperture 1mm by 3mm, each graph and two-dimensional simulation show a prominent bright maxima to either side of a dimmer central maxima that is completely absent with a wave of 600nm (Figure 7c). All three intensity decay plots show clearly dimmer fringes at the very edge that are otherwise hard to see in the intensity profile graphs and two-dimensional diffraction simulation.

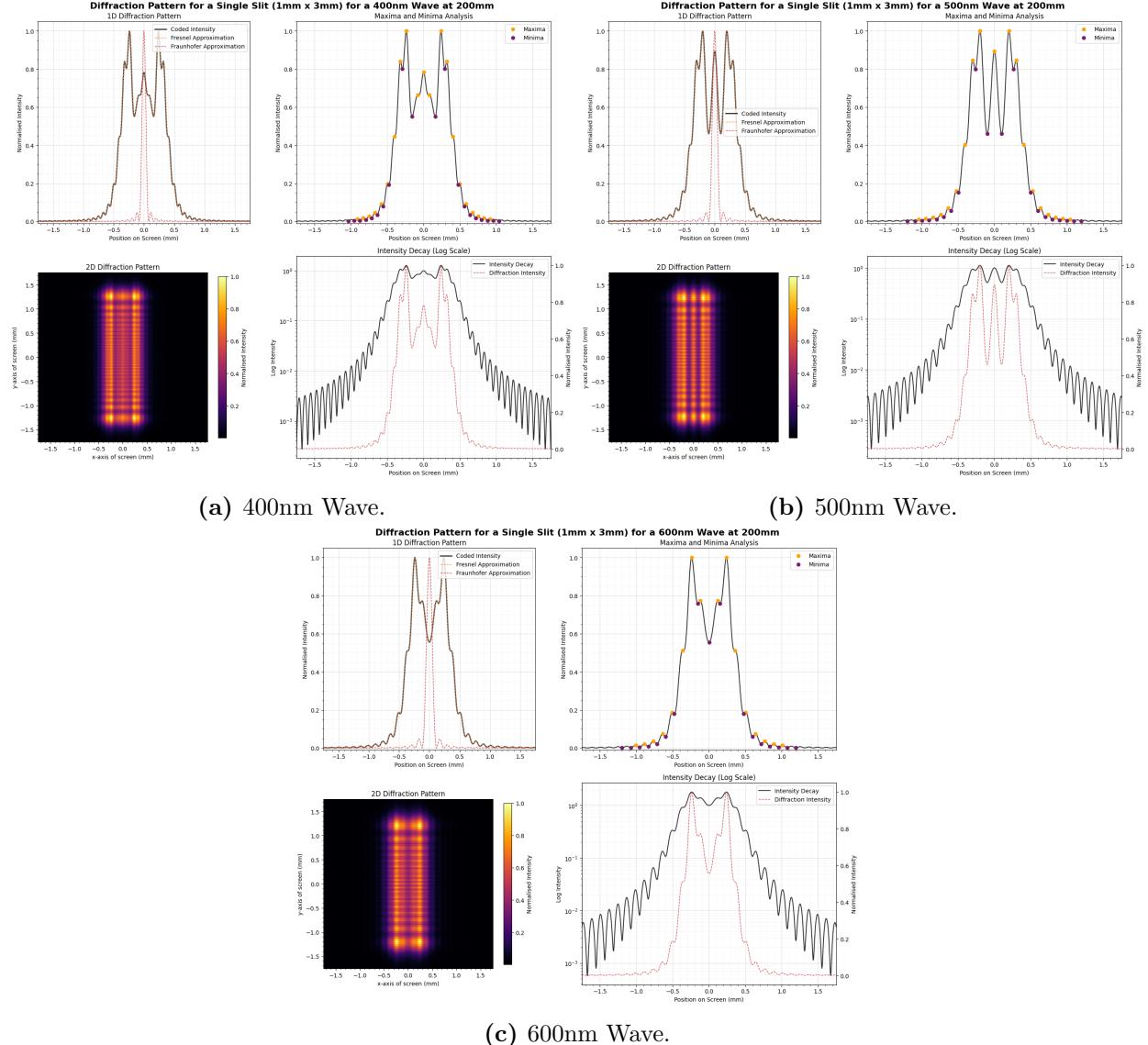


Figure 7: Diffraction Pattern for a Single Slit (1mm x 3mm) for a (a) 400nm; (b) 500nm; (c) 600nm Wave at 200mm.

The most visible and intense central maximum is seen with the 500nm wave (Figure 7b), present also with the 400nm wave but less intensely (Figure 7a). The two-dimensional renders of the diffraction patterns appear to have an airy look to them due to the fact that the fringes are closely spaced together.

4.1.2 Vertical Aperture — Varying Distance to Screen

Keeping the aperture dimensions as 1mm by 3mm and considering the same wavelengths as before, varying the distance +100mm and -100mm from the previous 200mm produces new diffraction patterns. The closer distance of 100mm generally created more fringes whilst the farther distance of 300mm divided the pattern into two more defined fringes as before with the 200mm.

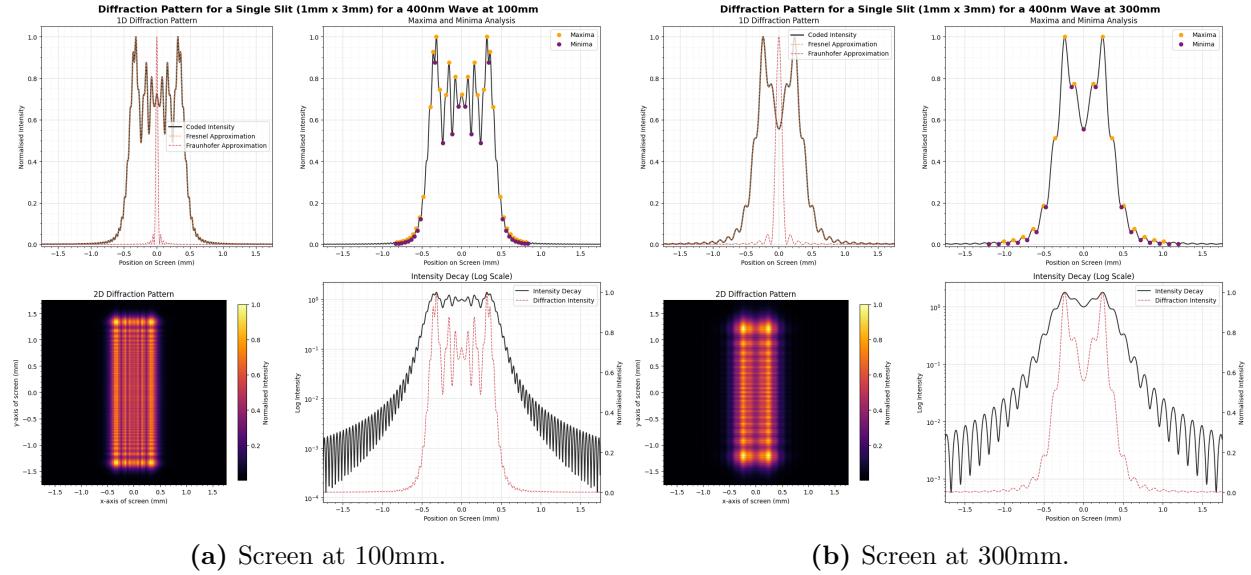


Figure 8: Diffraction Pattern for a Single Slit (1mm x 3mm) for a 400nm Wave at (a) 100mm and (b) 300mm.

At 100mm for a 400nm wave (Figure 8a) there appears to be more poorly-defined fringes closely packed together with the brightest fringes remaining in the outmost edges. In contrast, at 300mm (Figure 8b) the fringes appear as only two but much better and clearly defined despite the 'airy' surrounding fringes.

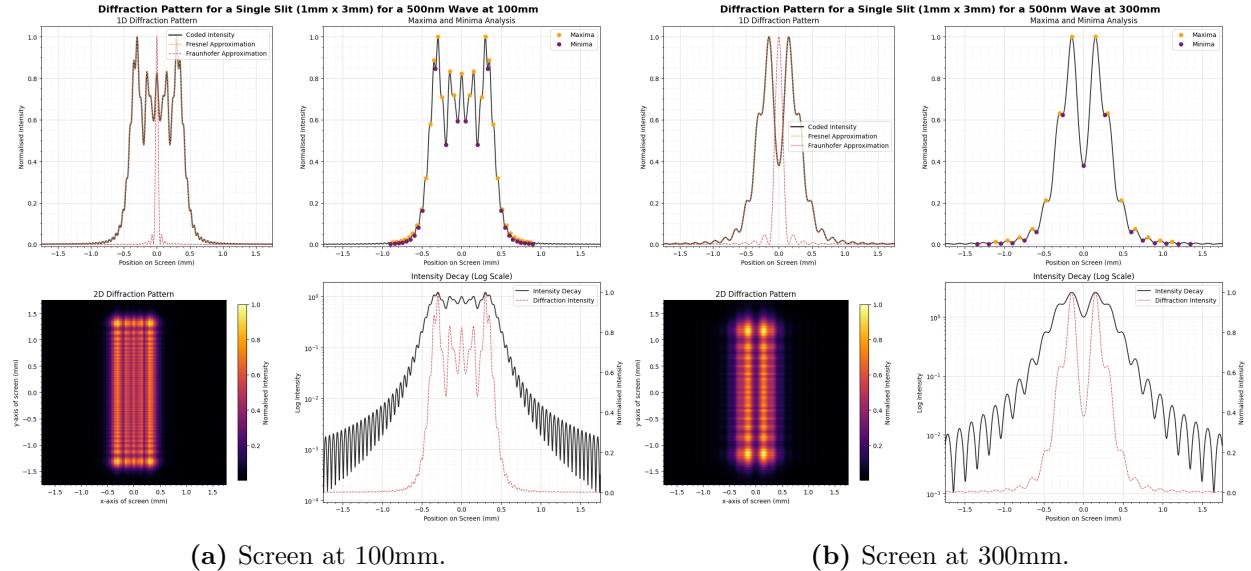


Figure 9: Diffraction Pattern for a Single Slit (1mm x 3mm) for a 500nm Wave at (a) 100mm and (b) 300mm.

The fringes at 100mm for a 500nm wave (Figure 9a) appear better than defined than those seen with the 400nm wave as less fringes appear, reducing the 'airy' look, the outermost fringes

remaining the brightest. At 300mm (Figure 9b) the two well-defined fringes observed for the 400nm wave appear closer together, remaining just as bright and clear with the same 'airy' look to the sides.

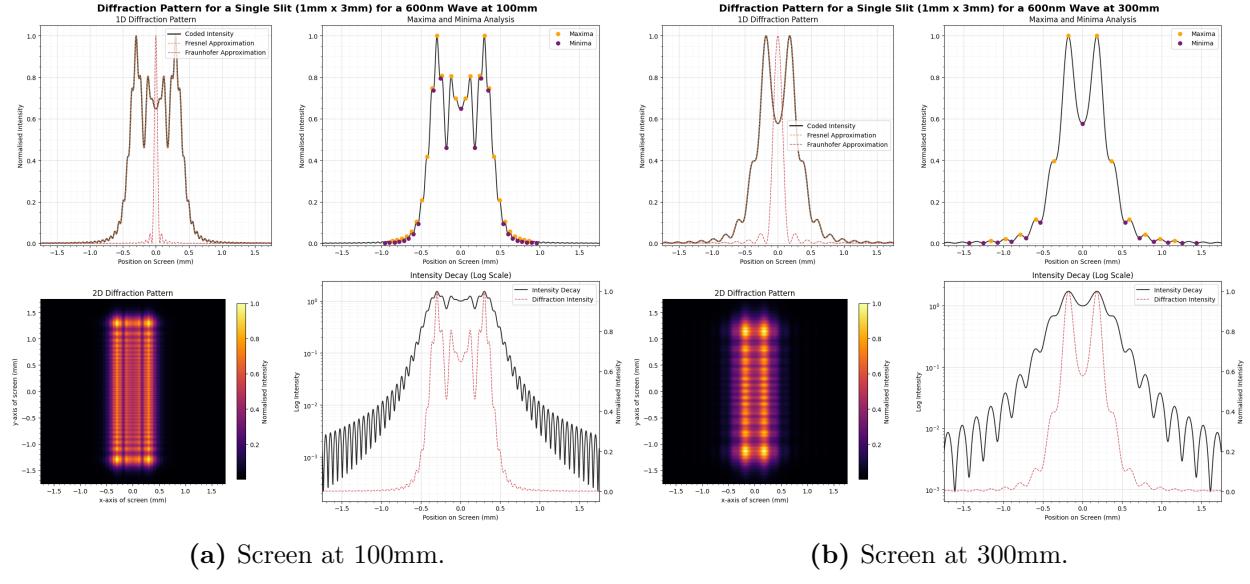
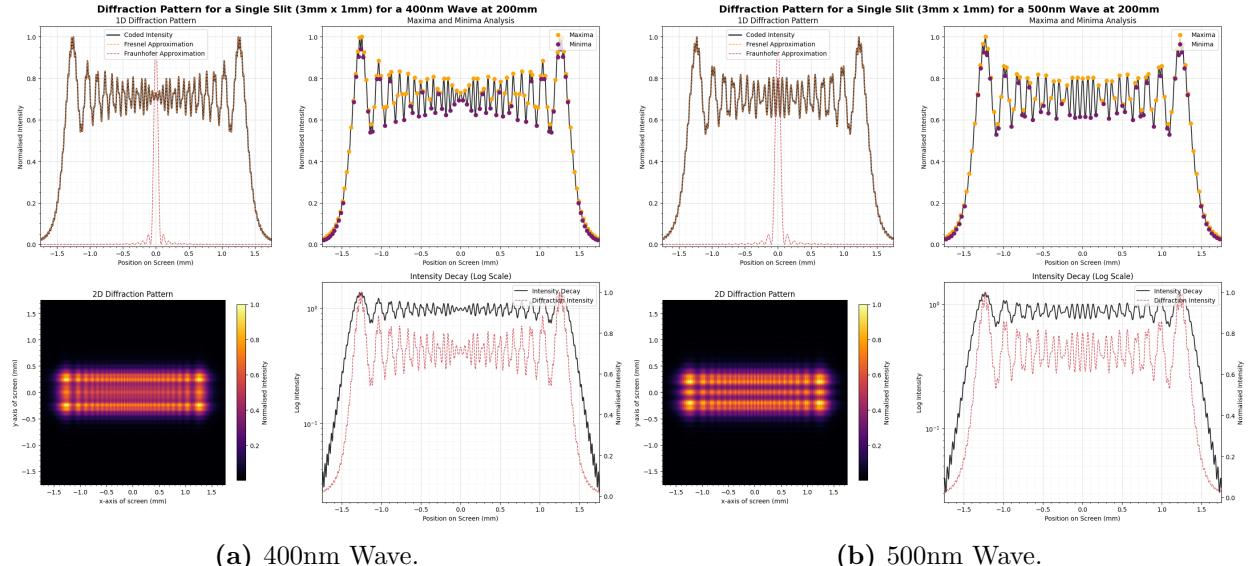


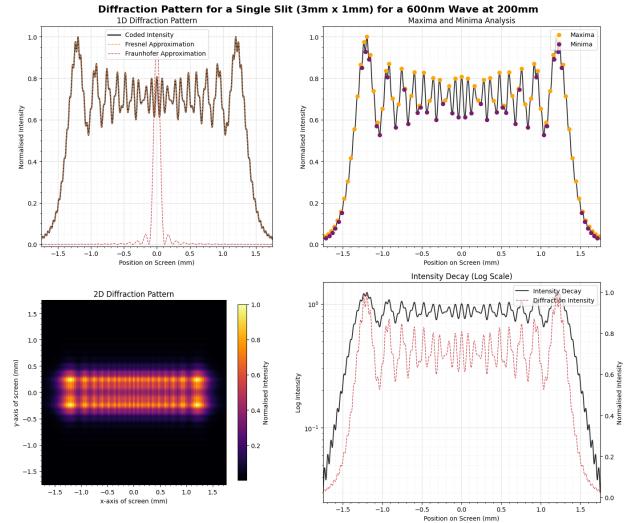
Figure 10: Diffraction Pattern for a Single Slit (1mm x 3mm) for a 600nm Wave at (a) 100mm and (b) 300mm.

For a 600nm wave projected onto a screen 100mm away (Figure 10a) the fringes are much less defined than previously, graphically showing there are four bright distinct fringes but visually appearing more like three due to the poorly-defined intensity difference between them, the two brightest fringes remaining as the outermost ones. At 300mm (Figure 10b) two distinct bright fringes remain the most well-defined with a less-defined diving minima between them, intensifying the 'airy' look.

4.1.3 Horizontal Aperture

Considering the two-dimensional simulated render of the diffraction pattern, what is observed is incredibly similar to the pattern visualised when the slit was vertical. Graphically, it is where most changes occur.





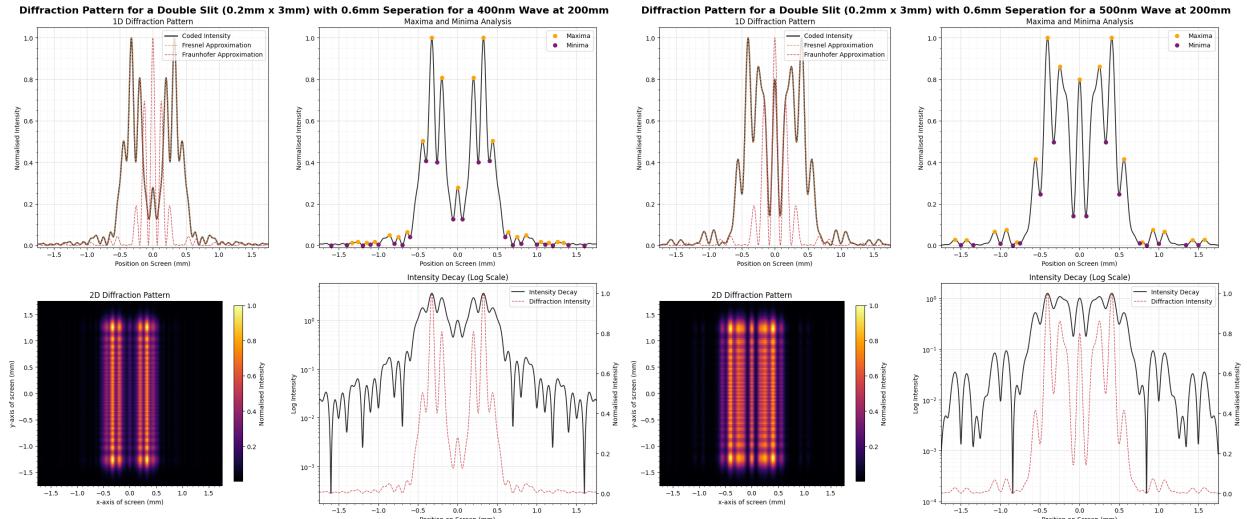
(c) 600nm Wave.

Figure 11: Diffraction Pattern for a (Horizontal) Single Slit (1mm x 3mm) for a (a) 400nm; (b) 500nm; (c) 600nm Wave at 200mm.

The graphs represented in Figure (11) above are all much wider than what was seen when the slit was vertical, the intensity profile representing the smaller fringes that make up the bigger, better-defined fringes that were observed in Figure (7). In all graphs the centre of the pattern is seen to be the most tightly-packed, the best-defined fringes at both ends of the pattern. The patterns continue to have the same 'airy' blurred look to them.

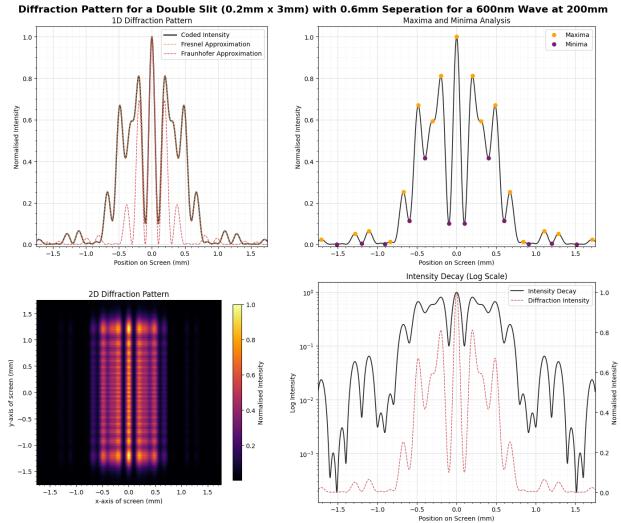
4.2 Rectangular Aperture — Double Slit

For the double slit diffraction pattern simulation renders two distinct bright fringes are always present, with a central fringe that is dimmest at 400nm (Figure 12a) and brightest at 600nm (Figure 12c). The two fringes to either side of the central fringe get broader and less-defined with increasing wavelength. Distinctly, two separate fringe groups (excluding central) can be observed for all three wavelengths.



(a) 400nm Wave.

(b) 500nm Wave.



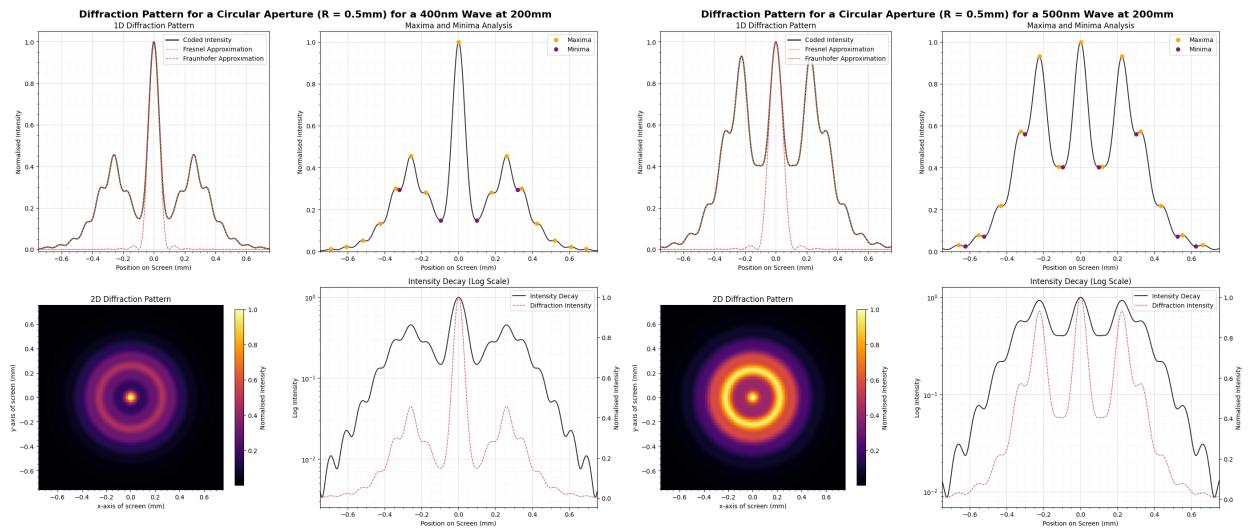
(c) 600nm Wave.

Figure 12: Diffraction Pattern for a Double Slit (0.2mm x 3mm) with 0.6mm Separation for a (a) 400nm; (b) 500nm; (c) 600nm Wave at 200mm.

Graphically a trend can be seen in the increasing intensity of the central fringe with increasing wavelength, surrounding fringes gaining a more 'airy' look to them the brighter the central fringe is.

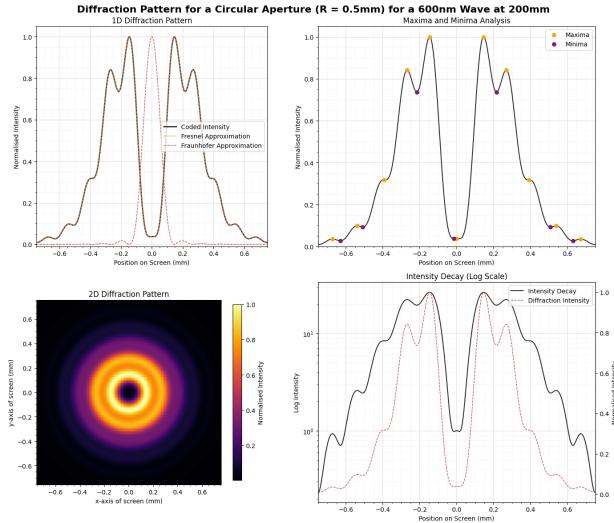
4.3 Circular Aperture

For a circular aperture the two-dimensional diffraction pattern render is very different for each, appearing to move outwards with increasing wavelengths. Wavelengths 400nm (Figure 13a) and 500nm (Figure 13b) have bright central maxima whilst for 600nm (Figure 13c) there is a central minimum instead. The 500nm and 600nm both are observed to have brighter surrounding ring fringes, the 400nm wave having quite weak fringes. All concentric fringes appear very well-defined with an 'airy' appearance for the outermost darkest ring.



(a) 400nm Wave.

(b) 500nm Wave.



(c) 600nm Wave.

Figure 13: Diffraction Pattern for a Circular Aperture ($R = 0.5\text{mm}$) for a (a) 400nm; (b) 500nm; (c) 600nm Wave at 200mm.

The graph for the pattern formed by the 400nm wave (Figure 13a) has a large and prominent central peak with a bright airy ring around it. The graph for the 500nm wave (Figure 13b) has a similar bright central maximum and the surrounding ring is of the same intensity as the central peak with 'airy' rings on the outside. The 600nm wave (Figure 13c) has a prominent, fully dark central minimum surrounded by a bright ring.

5 Analysis & Discussion

The diffraction patterns obtained from computational simulation can be linked and interpreted through the theoretical context outlined in §2. By applying the Huygens-Fresnel Principle to each modelled apertures a collection of secondary point sources was computationally reproduced to emit spherical wavelets (§2.1–2.2). Theoretical framework focused mostly on the Fraunhofer (far-field) diffraction patterns for the different apertures (§2.3.2–2.3.4), but the electric field equation used in the coding implementation (§2.1.1, Eq. (4)) most closely relates to the Huygens-Fresnel integral (Eq. (7)) due to the fact that the parameters were set in the Fresnel (near-field) regime. As a result, the simulations most resemble the near-field behaviour which were then compared to the analytical Fraunhofer expressions (Eq. (6), (8), (9)) graphically.

5.1 Single Slit Diffraction

The results observed for the rectangular single slit (Figures 7–11) reproduce the characteristic intensity profile described in §2.3.2: a dominant central maximum with diminishing side fringes. The trend expected with the diffraction pattern presents itself in the plots with increasing wavelengths producing wider patterns consistent with the theoretical context. However, the computed profile of the intensity did not correspond with the sinc^2 form typically expected. This is due to the fact that this form is only present in the far-field regime. Instead, for all graphs, the side lobes appear broader and the minima less sharply defined, giving the appearance of a "doubled" sinc^2 form.

This discrepancy can then be said to arise due to the fact that the Fraunhofer expressions assume far-field, whereas the computational approach retained the full Fresnel description and

Huygens assumptions. At the simulated distances observed (100mm–300mm) the curvature of the spherical wavelets was still quite significant due to the proximity of the primary source to the screen, so the expected destructive interference at the first minima was incomplete and therefore not observed. In the Fraunhofer approximation the intensity at the first minima falls directly to zero, and for the computed curves the minima was found to be non-zero and smoother side fringes that were visually interpreted as an ‘airy’ look. This effect was more pronounced at the shorter distance (100mm) with the fringes becoming less distinct (Figures 8a–10a), while in the intensity profile graphs for the wavelengths at 300mm (Figures 8b–10b), the distinct double fringe pattern observed appears closer and more defined as it begins entering the Fraunhofer field.

5.2 Double Slit Diffraction

The results for the double slit (Figure 12) observed showcased the expected combination of diffraction and interference described in §2.3.3: a set of finer fringes contained within the broader diffraction envelope of the single slit. As with the single slit, the spacing and intensity of the fringes was dependant on the wavelength of the incident light, with the longer wavelengths producing broader envelopes and less distinct fringes. As was observed for the single slit, the patterns did not correspond to the ideal Fraunhofer form that was expected with Equation (8) as the minima were non-zero and the central peak diffraction was wider than the formula predicted.

The reasons for these discrepancies are parallel to the discrepancies found with the single slit. Due to the use of the full Huygens-Fresnel integral in the computed diffraction patterns the contributions of the Fresnel aspect of the formula over the finite screen distance meant that the destructive interference was incomplete and so the expected minima at zero that would be observed with the Fraunhofer approach were not present. In the intensity profile that was rendered the fringes appear broadened and ‘blurred’ though never fully cancelled. For the longest wavelength used in the computation (600nm) the intensity profile obtained appears most closely related to the Fraunhofer approximation (Figure 12c), showing a relationship between the distance to the screen and the wavelength.

5.3 Circular Aperture Diffraction

The computed circular aperture results (Figure 13) reproduced the expected airy-pattern of concentric circular fringes that was outlined in §2.3.4. A bright central maxima was observed for the pattern produced by the 400nm and 500nm wavelengths, surrounded by concentric rings of decreasing intensity consistent with the Bessel function dependance observed in Equation (9). At 600nm a central intensity minimum is observed instead surrounded by bright concentric rings formed by altering constructive and destructive interference to form a pattern.

The Fresnel features observed for the single and double slits are also present with the circular aperture, the outer rings appearing blurred with incomplete minima and reduced contrasting intensity that would be otherwise present with the Fraunhofer pattern. The semi-logarithmic plots of the intensity decay showcase the predicted rapid fall-off of the ring intensity that produces the ‘airy’ appearance. The results obtained show that the Fraunhofer airy disc pattern is an approximation and that, at finite distances, the Fresnel effects alter both the clarity and depth of the produced rings.

6 Conclusion

The computational study successfully showcased and modelled the diffraction patterns obtained from single and double rectangular apertures, as well as circular apertures through the use of the Huygens-Fresnel Principle. The results reproduced the key features that were predicted and discussed in the theory section, including the central maxima, an encasing envelope that broadened with wavelength, and the characteristic interference fringes. The equations used to compare to the Fraunhofer approximation of the expected diffraction patterns in §3 did not align with the computed pattern as predicted, reflecting the fact that the computational renders retained the Fresnel properties of near-field diffraction. At greater distances from the screen, the intensity profiles increasingly resembled the Fraunhofer forms, confirming them as far-field approximations contained within the Huygens-Fresnel framework.

Overall, the study verified the theoretical foundations of wave optics regarding diffraction and interference patterns, while also displaying the distinction between the Fraunhofer (far-field) and Fresnel (near-field) regimes with consideration to the patterns produced. The main limitations came from the approach to modelling the aperture grid, intensity normalisation, and the finite distance to the screen. These factors collectively reduced the contrast and clarity of the diffraction patterns, blurring the resultant fringes and dark minima. For future renditions of this computational experiment, a finer aperture sampling size, modelling larger propagations distances for comparison, and exploring light that isn't coherent and monochromatic could be used.

References

- [1] M. Born, E. Wolf, A. B. Bhatia, P. C. Clemmow, D. Gabor, A. R. Stokes, A. M. Taylor, P. A. Wayman, and W. L. Wilcock, *Principles of Optics: Electromagnetic Theory of Propagation, Interference and Diffraction of Light*, 7th ed. Cambridge University Press, 1999.
- [2] E. Hecht, *Optics*. Pearson Education India, 2012.
- [3] O. et al., *University Physics Volume 3*. OpenStax, 2016. [Online]. Available: "<https://openstax.org/details/books/university-physics-volume-3>"
- [4] UCD, “7. Diffraction Pattern due to a Rectangular Aperture,” *3rd Year Astro and Space Physics Laboratory Manual*, n.d.
- [5] Wikipedia contributors, “Huygens–fresnel principle — Wikipedia, the free encyclopedia,” 2025, [Accessed 27-September-2025]. [Online]. Available: https://en.wikipedia.org/w/index.php?title=Huygens%20%93Fresnel_principle&oldid=1291861847
- [6] K. K. Likharev, “Essential graduate physics,” *EM: Classical Electrodynamics, part*, vol. 7, 2013.
- [7] Amoli, “Huygens Principle for a Near Source and Far Off Source Diagram,” 2019, [Accessed 27-September-2025]. [Online]. Available: "<https://www.sarthaks.com/254292/verify-laws-of-reflection-or-laws-of-refraction-on-the-basis-of-huygens-wave-theory>"
- [8] J. M. Cowley, *Diffraction physics*. Elsevier, 1995.
- [9] M. Norouzpour, “The Development of Self-interference of Split HOLZ (SIS-HOLZ) lines for Measuring z-dependent Atomic Displacement in Crystals,” Ph.D. dissertation, University of Tehran, 04 2017.
- [10] Wikipedia contributors, “Diffraction — Wikipedia, the free encyclopedia,” 2025, [Accessed 27-September-2025]. [Online]. Available: <https://en.wikipedia.org/w/index.php?title=Diffraction&oldid=1313573322>
- [11] J. I. Thomas, “The Classical Double Slit Interference Experiment: A New Geometrical Approach,” *American Journal of Optics and Photonics*, vol. 7, no. 1, pp. 1–9, 2019. [Online]. Available: <https://doi.org/10.11648/j.ajop.20190701.11>
- [12] C. Andrews, “Diffraction pattern of a circular aperture at short distances,” *Physical Review*, vol. 71, no. 11, p. 777, 1947.
- [13] D. Burch, “Fresnel diffraction by a circular aperture,” *American Journal of Physics*, vol. 53, no. 3, pp. 255–260, 1985.
- [14] E. Koushki and S. Alavi, “Diffraction of a partial temporal coherent beam from a single-slit and a circular aperture,” *Optics Communications*, vol. 441, pp. 33–37, 2019.
- [15] I. Cooper and D. Arora, “Single-slit diffraction — visual physics,” n.d., [Accessed 27-September-2025]. [Online]. Available: "https://d-arora.github.io/VisualPhysics/mod31/m31_singleSlit.htm"
- [16] Wikipedia contributors, “Fresnel diffraction — Wikipedia, the free encyclopedia,” 2025, [Accessed 3-October-2025]. [Online]. Available: https://en.wikipedia.org/w/index.php?title=Fresnel_diffraction&oldid=1314266286

- [17] S. J. Ling, “Physics Bootcamp,” 2025, [Accessed 3-October-2025]. [Online]. Available: [“https://www.physicsbootcamp.org/author-bio-TWJ.html”](https://www.physicsbootcamp.org/author-bio-TWJ.html)
- [18] Anon., “13. Lecture, 14 October 1999,” 1999, [Accessed 3-October-2025]. [Online]. Available: [“https://www.pas.rochester.edu/~dmw/ast203/Lectures/Lect_13.pdf”](https://www.pas.rochester.edu/~dmw/ast203/Lectures/Lect_13.pdf)

List of Figures

1	Visual representation of a spherical wavefront from a point source; graph illustrates the amplitude decreasing with distance as the wave propagates radially outward. [2]	2
2	Diagram of Huygens’ Principle for a near source (left) and far off source (right), showcasing the secondary wavelets at the secondary wavefront. [7]	3
3	Intensity profiles in the near-field diffraction (middle) compared with the far-field diffraction (right) intensity profiles. [9]	4
4	Single slit diffraction diagram; infinitely many points (three) shown along length d project phase contributions from the wavefront, producing a continuously varying intensity θ on the registering plate. [10]	4
5	Double slit diffraction diagram; intensity falls off with angle to form an interference pattern consisting of a series of bright and dark fringes. [3]	5
6	Circular aperture diffraction pattern; Fraunhofer (far-field) diffraction (top), Fresnel (near-field) diffraction (bottom) where a central black spot can be observed. [15]	5
7	Diffraction Pattern for a Single Slit (1mm x 3mm) for a (a) 400nm; (b) 500nm; (c) 600nm Wave at 200mm.	8
8	Diffraction Pattern for a Single Slit (1mm x 3mm) for a 400nm Wave at (a) 100mm and (b) 300mm.	9
9	Diffraction Pattern for a Single Slit (1mm x 3mm) for a 500nm Wave at (a) 100mm and (b) 300mm.	9
10	Diffraction Pattern for a Single Slit (1mm x 3mm) for a 600nm Wave at (a) 100mm and (b) 300mm.	10
11	Diffraction Pattern for a (Horizontal) Single Slit (1mm x 3mm) for a (a) 400nm; (b) 500nm; (c) 600nm Wave at 200mm.	11
12	Diffraction Pattern for a Double Slit (0.2mm x 3mm) with 0.6mm Separation for a (a) 400nm; (b) 500nm; (c) 600nm Wave at 200mm.	12
13	Diffraction Pattern for a Circular Aperture ($R = 0.5\text{mm}$) for a (a) 400nm; (b) 500nm; (c) 600nm Wave at 200mm.	13

Appendix — Code

```
1 # %%
2 import numpy as np
3 import matplotlib.pyplot as plt
4 import matplotlib.colors
5 from scipy.signal import find_peaks
6 from scipy.special import j1
7
8 # %%
9 lam400 = '#8200b5' # Hex color code for 400nm wavelength (violet)
10 lam500 = '#00ff92' # Hex color code for 500nm wavelength (greenish)
11 lam600 = '#ffbe00' # Hex color code for 600nm wavelength (yellow)
12
13 # https://academo.org/demos/wavelength-to-colour-relationship/
14
15 # %% [markdown]
16 # Vertical Single Slit
17
18 # %%
19 # Wave Equation and Properties
20
21 w, h = 1e-3, 3e-3
22 L = 200e-3 # Distance from aperture to screen
23 lam = 500e-9 # Wavelengths of light
24
25 k = 2 * np.pi / lam # Wave number/vector
26
27 # Defining Aperture
28 nx, ny = 300, 900
29 x_ap = np.linspace(-w/2, w/2, nx)
30 y_ap = np.linspace(-h/2, h/2, ny)
31 xx_ap, yy_ap = np.meshgrid(x_ap, y_ap)
32
33 dx = x_ap[1] - x_ap[0]
34 dy = y_ap[1] - y_ap[0] # Spacing between points in the aperture grid
35 area = dx * dy # Area of each point source in the aperture (Riemann sum element)
36
37 mask = (np.abs(xx_ap) <= w/2) & (np.abs(yy_ap) <= h/2)
38
39 xx_ap_srect = xx_ap[mask].ravel()
40 yy_ap_srect = yy_ap[mask].ravel() # Flattening the arrays to 1D for easier computation
41
42 aperture_points = np.stack((xx_ap_srect, yy_ap_srect), axis=-1) # Creating a list of (x,y) points in the aperture
43
44 # Defining the Wave Equation Functions for 1D and 2D
45
46 def srect_1d(screen=(-1.75e-3, 1.75e-3), points=1000):
47
48     screen = np.linspace(screen[0], screen[1], points) # Screen
49
50     I = [] # List to hold the resultant field values at each point on the screen (intensity)
51     for x in screen:
52         r = np.sqrt((x - aperture_points[:,0])**2 + (aperture_points[:,1])**2 + L**2)
53         E = (np.exp(1j * k * r) / r) * area # Electric field at each point on the screen due to each point in the aperture
54         E_total = E.sum() # Total electric field
55         I.append(np.abs(E_total)**2)
56
57     I = np.array(I) # Convert list to array for easier plotting
58     I /= I.max() # Normalize intensity for plotting
59
60     return screen, I
61
62 def fraun_srect_1d(screen=(-1.75e-3, 1.75e-3), points=1000):
63
64     screen = np.linspace(screen[0], screen[1], points)
65
66     theta = np.arctan(screen / L)
67     beta = (np.pi * w / lam) * np.sin(theta)
68
69     I_fraun = (np.sinc(beta/np.pi)**2)
70     I_fraun /= I_fraun.max()
71
72     return screen, I_fraun
```

```

73
74 def fresnel_srect_1d(screen=(-1.75e-3, 1.75e-3), points=1000):
75
76     screen = np.linspace(screen[0], screen[1], points)
77     factor = (np.exp(1j * k * L)) / (1j * lam * L) # Fresnel Approximation
78
79     xp = aperture_points[:,0]
80     yp = aperture_points[:,1]
81
82     E_theory = np.zeros_like(screen, dtype=complex)
83     for i, x in enumerate(screen):
84         phase = np.exp((1j * k)/(2 * L) * ((x - xp)**2 + yp**2)) # Fresnel Approximation ( Integral Part)
85         E = phase * area
86         E_theory[i] = factor * E.sum()
87
88     I_fresnel = (np.abs(E_theory)**2)
89     I_fresnel /= I_fresnel.max()
90
91     return screen, I_fresnel
92
93 def srect_2d(screen=(-1.75e-3, 1.75e-3), screen_res=1.25e-5):
94
95     screen_x1, screen_x2 = screen
96     screen_y1, screen_y2 = screen
97     n = int((screen_x2 - screen_x1) / screen_res)
98     m = int((screen_y2 - screen_y1) / screen_res)
99
100    X = np.linspace(screen_x1, screen_x2, n)
101    Y = np.linspace(screen_y1, screen_y2, m)
102    X, Y = np.meshgrid(X, Y) # 2D grid of screen points
103
104    E_total = np.zeros((m, n), dtype=complex)
105    for ap in aperture_points:
106        r = np.sqrt((X - ap[0])**2 + (Y - ap[1])**2 + L**2) # ap[0] and ap[1] are the x and y coordinates of the aperture point
107        E_total += (np.exp(1j * k * r) / r) * area # Sum contributions from each point source
108
109    I = np.abs(E_total)**2
110    I /= I.max()
111
112    return X, Y, I
113
114 # Patterns
115 screen, I1D = srect_1d()
116 screen, I1DF = fresnel_srect_1d()
117 screen, I1DFh = fraun_srect_1d()
118 X, Y, I2D = srect_2d()
119
120 # Plots
121 fig, axes = plt.subplots(2, 2, figsize=(14, 12), gridspec_kw={'width_ratios':[1, 1.2]})
```

fig.suptitle('Diffraction Pattern for a Single Slit (1mm x 3mm) for a 500nm Wave at 200mm',
 fontsize=16, fontweight='bold')

```

122
123 # 1D Plot; top left
124 axes[0,0].plot(screen * 1e3, I1D, color='#2A2A2A', linewidth=2, label='Coded Intensity')
125 axes[0,0].plot(screen * 1e3, I1DF, '--', color="#f68d45", linewidth=1, label='Fresnel Approximation')
126 axes[0,0].plot(screen * 1e3, I1DFh, '--', color="#cb4149", linewidth=1, label='Fraunhofer Approximation')
127
128 axes[0,0].set_title('1D Diffraction Pattern')
129 axes[0,0].set_xlabel('Position on Screen (mm)')
130 axes[0,0].set_ylabel('Normalised Intensity')
131 axes[0,0].set_ylim([-0.01, 1.05])
132 axes[0,0].set_xlim([-1.75, 1.75])
133
134 axes[0,0].minorticks_on()
135 axes[0,0].grid(True, which='major', linewidth=0.8, color='#DDDDDD', zorder=2)
136 axes[0,0].grid(True, which='minor', linewidth=0.5, color='#EEEEEE', linestyle='--', zorder=1)
137 axes[0,0].legend()
138
139 # 2D Plotl; bottom left
140 graph = axes[1,0].pcolormesh(X*1e3, Y*1e3, I2D, shading='auto', cmap='inferno')
141 axes[1,0].set_aspect('equal')
```

```

144 axes[1,0].set_title('2D Diffraction Pattern')
145 axes[1,0].set_xlabel('x-axis of screen (mm)')
146 axes[1,0].set_ylabel('y-axis of screen (mm)')
147 fig.colorbar(graph, ax=axes[1,0], orientation='vertical', label='Normalised Intensity', shrink
    =0.8)
148 axes[1,0].minorticks_on()
149
150 # Peak and Valley Slices; top right
151 # https://docs.scipy.org/doc/scipy/reference/generated/scipy.signal.find_peaks.html#scipy.
    signal.find_peaks
152 peaks, _ = find_peaks(I1D, height=0.01)
153 valleys, _ = find_peaks(-I1D, prominence=0.005)
154
155 axes[0,1].plot(screen * 1e3, I1D, color='#2A2A2A', linewidth=1.5)
156 axes[0,1].plot(screen[peaks] * 1e3, I1D[peaks], 'o', color='#fca50a', label='Maxima')
157 axes[0,1].plot(screen[valleys] * 1e3, I1D[valleys], 'o', color="#781c6d", label='Minima')
158 axes[0,1].set_title('Maxima and Minima Analysis')
159 axes[0,1].set_xlabel('Position on Screen (mm)')
160 axes[0,1].set_ylabel('Normalised Intensity')
161 axes[0,1].set_ylim([-0.01, 1.05])
162 axes[0,1].set_xlim([-1.75, 1.75])
163
164 axes[0,1].grid(True, which='major', linewidth=0.8, color='#DDDDDD', zorder=2)
165 axes[0,1].grid(True, which='minor', linewidth=0.5, color='#EEEEEE', linestyle='--', zorder=1)
166 axes[0,1].minorticks_on()
167 axes[0,1].legend(loc='upper right')
168
169 # Intensity Decay Log Plot; bottom right
170 centre = np.argmin(np.abs(screen))
171 centre_peak = I1D[centre]
172
173 I1D_norm = I1D / centre_peak
174
175 mask = I1D_norm > 1e-12
176
177 ax_log = axes[1,1]
178
179 ax_log.semilogy(screen * 1e3, I1D_norm[mask], color='#2A2A2A', linewidth=1.5, label='Intensity
    Decay')
180 ax_log.set_ylabel('Log Intensity')
181 ax_log.tick_params(axis='y')
182
183 ax_lin = ax_log.twinx()
184 ax_lin.plot(screen * 1e3, I1D, '--', color='#cb4149', linewidth=1, label='Diffraction
    Intensity')
185 ax_lin.set_ylabel('Normalised Intensity')
186 ax_lin.tick_params(axis='y')
187
188 ax_log.set_xlabel('Position on Screen (mm)')
189 ax_log.set_title('Intensity Decay (Log Scale)')
190 ax_log.set_xlim([-1.75, 1.75])
191
192 ax_log.grid(True, which='major', linewidth=0.8, color='#DDDDDD', zorder=2)
193 ax_log.grid(True, which='minor', linewidth=0.5, color='#EEEEEE', linestyle='--', zorder=1)
194 ax_log.minorticks_on()
195
196 lines1, labels1 = ax_log.get_legend_handles_labels()
197 lines2, labels2 = ax_lin.get_legend_handles_labels()
198 ax_log.legend(lines1 + lines2, labels1 + labels2, loc='upper right')
199
200 plt.tight_layout()
201 plt.subplots_adjust(wspace=0.2, top=0.94)
202
203 plt.show()
204
205 # %% [markdown]
206 # # Horizontal Single Slit
207
208 # %%
209 # Wave Equation and Properties
210
211 w, h = 3e-3, 1e-3
212 L = 200e-3 # Distance from aperture to screen
213 lam = 500e-9 # Wavelengths of light
214
215 k = 2 * np.pi / lam # Wave number/vector

```

```

216
217 # Defining Aperture
218 nx, ny = 300, 900
219 x_ap = np.linspace(-w/2, w/2, nx)
220 y_ap = np.linspace(-h/2, h/2, ny)
221 xx_ap, yy_ap = np.meshgrid(x_ap, y_ap)
222
223 dx = x_ap[1] - x_ap[0]
224 dy = y_ap[1] - y_ap[0] # Spacing between points in the aperture grid
225 area = dx * dy # Area of each point source in the aperture (Riemann sum element)
226
227 mask = (np.abs(xx_ap) <= w/2) & (np.abs(yy_ap) <= h/2)
228
229 xx_ap_srect = xx_ap[mask].ravel()
230 yy_ap_srect = yy_ap[mask].ravel() # Flattening the arrays to 1D for easier computation
231
232 aperture_points = np.stack((xx_ap_srect, yy_ap_srect), axis=-1) # Creating a list of (x,y)
233 points in the aperture
234
235 # Defining the Wave Equation Functions for 1D and 2D
236
237 def srect_1d(screen=(-1.75e-3, 1.75e-3), points=1000):
238
239     screen = np.linspace(screen[0], screen[1], points) # Screen
240
241     I = [] # List to hold the resultant field values at each point on the screen (intensity)
242     for x in screen:
243         r = np.sqrt((x - aperture_points[:,0])**2 + (aperture_points[:,1])**2 + L**2)
244         E = (np.exp(1j * k * r) / r) * area # Electric field at each point on the screen due
245         to each point in the aperture
246         E_total = E.sum() # Total electric field
247         I.append(np.abs(E_total)**2)
248
249     I = np.array(I) # Convert list to array for easier plotting
250     I /= I.max() # Normalize intensity for plotting
251
252     return screen, I
253
254 def fraun_srect_1d(screen=(-1.75e-3, 1.75e-3), points=1000):
255
256     screen = np.linspace(screen[0], screen[1], points)
257
258     theta = np.arctan(screen / L)
259     beta = (np.pi * h / lam) * np.sin(theta)
260
261     I_fraun = (np.sinc(beta/np.pi)**2)
262     I_fraun /= I_fraun.max()
263
264     return screen, I_fraun
265
266 def fresnel_srect_1d(screen=(-1.75e-3, 1.75e-3), points=1000):
267
268     screen = np.linspace(screen[0], screen[1], points)
269     factor = (np.exp(1j * k * L)) / (1j * lam * L) # Fresnel Approximation
270
271     xp = aperture_points[:,0]
272     yp = aperture_points[:,1]
273
274     E_theory = np.zeros_like(screen, dtype=complex)
275     for i, x in enumerate(screen):
276         phase = np.exp((1j * k)/(2 * L) * ((x - xp)**2 + yp**2)) # Fresnel Approximation (
277         Integral Part)
278         E = phase * area
279         E_theory[i] = factor * E.sum()
280
281     I_fresnel = (np.abs(E_theory)**2)
282     I_fresnel /= I_fresnel.max()
283
284     return screen, I_fresnel
285
286 def srect_2d(screen=(-1.75e-3, 1.75e-3), screen_res=1.25e-5):
287
288     screen_x1, screen_x2 = screen
289     screen_y1, screen_y2 = screen
290     n = int((screen_x2 - screen_x1) / screen_res)
291     m = int((screen_y2 - screen_y1) / screen_res)

```

```

289
290     X = np.linspace(screen_x1, screen_x2, n)
291     Y = np.linspace(screen_y1, screen_y2, m)
292     X, Y = np.meshgrid(X, Y) # 2D grid of screen points
293
294     E_total = np.zeros((m, n), dtype=complex)
295     for ap in aperture_points:
296         r = np.sqrt((X - ap[0])**2 + (Y - ap[1])**2 + L**2) # ap[0] and ap[1] are the x and y
297         coordinates of the aperture point
298         E_total += (np.exp(1j * k * r) / r) * area # Sum contributions from each point source
299
300     I = np.abs(E_total)**2
301     I /= I.max()
302
303     return X, Y, I
304
305 # Patterns
306 screen, I1D = srect_1d()
307 screen, I1DF = fresnel_srect_1d()
308 screen, I1DFh = fraun_srect_1d()
309 X, Y, I2D = srect_2d()
310
311 # Plots
312 fig, axes = plt.subplots(2, 2, figsize=(14, 12), gridspec_kw={'width_ratios':[1, 1.2]})
313
314 fig.suptitle('Diffraction Pattern for a Single Slit (3mm x 1mm) for a 500nm Wave at 200mm',
315               fontsize=16, fontweight='bold')
316
317 # 1D Plot; top left
318 axes[0,0].plot(screen * 1e3, I1D, color='#2A2A2A', linewidth=2, label='Coded Intensity')
319 axes[0,0].plot(screen * 1e3, I1DF, '--', color='#f68d45', linewidth=1, label='Fresnel
320   Approximation')
321 axes[0,0].plot(screen * 1e3, I1DFh, '--', color='#cb4149', linewidth=1, label='Fraunhofer
322   Approximation')
323
324 axes[0,0].set_title('1D Diffraction Pattern')
325 axes[0,0].set_xlabel('Position on Screen (mm)')
326 axes[0,0].set_ylabel('Normalised Intensity')
327 axes[0,0].set_xlim([-1.75, 1.75])
328
329 axes[0,0].minorticks_on()
330 axes[0,0].grid(True, which='major', linewidth=0.8, color="#DDDDDD", zorder=2)
331 axes[0,0].grid(True, which='minor', linewidth=0.5, color="#EEEEEE", linestyle='--', zorder=1)
332 axes[0,0].legend(loc='upper center')
333
334 # 2D Plotl; bottom left
335 graph = axes[1,0].pcolormesh(X*1e3, Y*1e3, I2D, shading='auto', cmap='inferno')
336 axes[1,0].set_aspect('equal')
337 axes[1,0].set_title('2D Diffraction Pattern')
338 axes[1,0].set_xlabel('x-axis of screen (mm)')
339 axes[1,0].set_ylabel('y-axis of screen (mm)')
340 fig.colorbar(graph, ax=axes[1,0], orientation='vertical', label='Normalised Intensity', shrink
341   =0.8)
342 axes[1,0].minorticks_on()
343
344 # Peak and Valley Slices; top right
345 # https://docs.scipy.org/doc/scipy/reference/generated/scipy.signal.find_peaks.html#scipy.
346   signal.find_peaks
347 peaks, _ = find_peaks(I1D, height=0.01)
348 valleys, _ = find_peaks(-I1D, prominence=0.005)
349
350 axes[0,1].plot(screen * 1e3, I1D, color='#2A2A2A', linewidth=1.5)
351 axes[0,1].plot(screen[peaks] * 1e3, I1D[peaks], 'o', color='#fca50a', label='Maxima')
352 axes[0,1].plot(screen[valleys] * 1e3, I1D[valleys], 'o', color='#781c6d', label='Minima')
353 axes[0,1].set_title('Maxima and Minima Analysis')
354 axes[0,1].set_xlabel('Position on Screen (mm)')
355 axes[0,1].set_ylabel('Normalised Intensity')
356 axes[0,1].set_xlim([-1.75, 1.75])
357
358 axes[0,1].grid(True, which='major', linewidth=0.8, color="#DDDDDD", zorder=2)
359 axes[0,1].grid(True, which='minor', linewidth=0.5, color="#EEEEEE", linestyle='--', zorder=1)
360 axes[0,1].minorticks_on()
361 axes[0,1].legend(loc='upper right')

```

```

359 # Intensity Decay Log Plot; bottom right
360 centre = np.argmin(np.abs(screen))
361 centre_peak = I1D[centre]
362
363 I1D_norm = I1D / centre_peak
364
365 mask = I1D_norm > 1e-12
366
367 ax_log = axes[1,1]
368
369 ax_log.semilogy(screen * 1e3, I1D_norm[mask], color="#2A2A2A", linewidth=1.5, label='Intensity
370 Decay')
371 ax_log.set_ylabel('Log Intensity')
372 ax_log.tick_params(axis='y')
373
374 ax_lin = ax_log.twinx()
375 ax_lin.plot(screen * 1e3, I1D, '--', color='#cb4149', linewidth=1, label='Diffraction
376 Intensity')
377 ax_lin.set_ylabel('Normalised Intensity')
378 ax_lin.tick_params(axis='y')
379
380 ax_log.set_xlabel('Position on Screen (mm)')
381 ax_log.set_title('Intensity Decay (Log Scale)')
382 ax_log.set_xlim([-1.75,1.75])
383
384 ax_log.grid(True, which='major', linewidth=0.8, color="#DDDDDD", zorder=2)
385 ax_log.grid(True, which='minor', linewidth=0.5, color="#EEEEEE", linestyle='--', zorder=1)
386 ax_log.minorticks_on()
387
388 lines1, labels1 = ax_log.get_legend_handles_labels()
389 lines2, labels2 = ax_lin.get_legend_handles_labels()
390 ax_log.legend(lines1 + lines2, labels1 + labels2, loc='upper right')
391
392 plt.tight_layout()
393 plt.subplots_adjust(wspace=0.2, top=0.94)
394
395 plt.show()
396
397 # %% [markdown]
398 # Double Slit
399
400 # Wave Equation and Properties
401 w, h = 0.2e-3, 3e-3
402 sep = 0.6e-3 # Distance between slits
403 L = 300e-3 # Distance from aperture to screen
404 lam = 500e-9 # Wavelengths of light
405
406 k = 2 * np.pi / lam # Wave number/vector
407
408 # Defining Aperture
409 nx, ny = 550, 1045 # Ratio ny = 1.9 * nx
410 x_ap = np.linspace(-(sep + w), (sep + w), nx)
411 y_ap = np.linspace(-h/2, h/2, ny)
412 xx_ap, yy_ap = np.meshgrid(x_ap, y_ap)
413
414 mask_s1 = (np.abs(xx_ap + sep/2) <= w/2) & (np.abs(yy_ap) <= h/2)
415 mask_s2 = (np.abs(xx_ap - sep/2) <= w/2) & (np.abs(yy_ap) <= h/2)
416 dmask = mask_s1 | mask_s2
417
418 xx_ap_drect = xx_ap[dmask].ravel()
419 yy_ap_drect = yy_ap[dmask].ravel()
420
421 dx = x_ap[1] - x_ap[0]
422 dy = y_ap[1] - y_ap[0] # Spacing between points in the aperture grid
423 area = dx * dy # Area of each point source in the aperture (Riemann sum element)
424
425 aperture_points = np.stack((xx_ap_drect, yy_ap_drect), axis=-1) # Creating a list of (x,y)
426 points in the aperture
427
428 # Defining the Wave Equation Functions for 1D and 2D
429
430 def drect_1d(screen=(-1.75e-3, 1.75e-3), points=1500):
431     screen = np.linspace(screen[0], screen[1], points) # Screen

```

```

432
433     I = [] # List to hold the resultant field values at each point on the screen (intensity)
434     for x in screen:
435         r = np.sqrt((x - aperture_points[:,0])**2 + (aperture_points[:,1])**2 + L**2)
436         E = (np.exp(1j * k * r) / r) * area # Electric field at each point on the screen due
437             to each point in the aperture
438         E_total = E.sum() # Total electric field
439         I.append(np.abs(E_total)**2)
440
441     I = np.array(I) # Convert list to array for easier plotting
442     I /= I.max() # Normalize intensity for plotting
443
444     return screen, I
445
446 def fraun_drect_1d(screen=(-1.75e-3, 1.75e-3), points=1500):
447
448     screen = np.linspace(screen[0], screen[1], points)
449
450     theta = np.arctan(screen / L)
451     beta = (np.pi * w / lam) * np.sin(theta)
452     alpha = (np.pi * sep / lam) * np.sin(theta)
453
454     I_fraun = (np.sinc(beta/np.pi)**2) * (np.cos(alpha)**2)
455     I_fraun /= I_fraun.max()
456
457     return screen, I_fraun
458
459 def fresnel_drect_1d(screen=(-1.75e-3, 1.75e-3), points=1500):
460
461     screen = np.linspace(screen[0], screen[1], points)
462     factor = (np.exp(1j * k * L)) / (1j * lam * L) # Fresnel Approximation
463
464     xp = aperture_points[:,0]
465     yp = aperture_points[:,1]
466
467     E_theory = np.zeros_like(screen, dtype=complex)
468     for i, x in enumerate(screen):
469         phase = np.exp((1j * k)/(2 * L) * ((x - xp)**2 + yp**2)) # Fresnel Approximation (
470             Integral Part)
471         E = phase * area
472         E_theory[i] = factor * E.sum()
473
474     I_fresnel = (np.abs(E_theory)**2)
475     I_fresnel /= I_fresnel.max()
476
477     return screen, I_fresnel
478
479 def drect_2d(screen=(-1.75e-3, 1.75e-3), screen_res=1.25e-5):
480
481     screen_x1, screen_x2 = screen
482     screen_y1, screen_y2 = screen
483     n = int((screen_x2 - screen_x1) / screen_res)
484     m = int((screen_y2 - screen_y1) / screen_res)
485
486     X = np.linspace(screen_x1, screen_x2, n)
487     Y = np.linspace(screen_y1, screen_y2, m)
488     X, Y = np.meshgrid(X, Y) # 2D grid of screen points
489
490     E_total = np.zeros((m, n), dtype=complex)
491     for ap in aperture_points:
492         r = np.sqrt((X - ap[0])**2 + (Y - ap[1])**2 + L**2) # ap[0] and ap[1] are the x and y
493             coordinates of the aperture point
494         E_total += (np.exp(1j * k * r) / r) * area # Sum contributions from each point source
495
496     I = np.abs(E_total)**2
497     I /= I.max()
498
499     return X, Y, I
500
501 # Patterns
502 screen, I1D = drect_1d()
503 screen, I1DF = fresnel_drect_1d()
504 screen, I1DFh = fraun_drect_1d()
505 X, Y, I2D = drect_2d()
506
507 # Plots

```

```

505 fig, axes = plt.subplots(2, 2, figsize=(14, 12), gridspec_kw={'width_ratios':[1, 1.2]})  

506  

507 fig.suptitle('Diffraction Pattern for a Double Slit (0.2mm x 3mm) with 0.6mm Separation for a  

508 500nm Wave at 300mm', fontsize=16, fontweight='bold')  

509  

510 # 1D Plot; top left  

511 axes[0,0].plot(screen * 1e3, I1D, color='#2A2A2A', linewidth=2, label='Coded Intensity')  

512 axes[0,0].plot(screen * 1e3, I1DF, '--', color="#f68d45", linewidth=1, label='Fresnel  

513 Approximation')  

514 axes[0,0].plot(screen * 1e3, I1DFh, '--', color='#cb4149', linewidth=1, label='Fraunhofer  

515 Approximation')  

516  

517 axes[0,0].set_title('1D Diffraction Pattern')  

518 axes[0,0].set_xlabel('Position on Screen (mm)')  

519 axes[0,0].set_ylabel('Normalised Intensity')  

520 axes[0,0].set_ylim([-0.01, 1.05])  

521 axes[0,0].set_xlim([-1.75, 1.75])  

522  

523 axes[0,0].minorticks_on()  

524 axes[0,0].grid(True, which='major', linewidth=0.8, color='#DDDDDD', zorder=2)  

525 axes[0,0].grid(True, which='minor', linewidth=0.5, color='#EEEEEE', linestyle='--', zorder=1)  

526 axes[0,0].legend()  

527  

528 # 2D Plot1; bottom left  

529 graph = axes[1,0].pcolormesh(X*1e3, Y*1e3, I2D, shading='auto', cmap='inferno')  

530 axes[1,0].set_aspect('equal')  

531 axes[1,0].set_title('2D Diffraction Pattern')  

532 axes[1,0].set_xlabel('x-axis of screen (mm)')  

533 axes[1,0].set_ylabel('y-axis of screen (mm)')  

534 fig.colorbar(graph, ax=axes[1,0], orientation='vertical', label='Normalised Intensity', shrink  

535 = 0.8)  

536 axes[1,0].minorticks_on()  

537  

538 # Peak and Valley Slices; top right  

539 # https://docs.scipy.org/doc/scipy/reference/generated/scipy.signal.find\_peaks.html#scipy.  

540 signal.find_peaks  

541 peaks, _ = find_peaks(I1D, height=0.01)  

542 valleys, _ = find_peaks(-I1D, prominence=0.005)  

543  

544 axes[0,1].plot(screen * 1e3, I1D, color='#2A2A2A', linewidth=1.5)  

545 axes[0,1].plot(screen[peaks] * 1e3, I1D[peaks], 'o', color='#fca50a', label='Maxima')  

546 axes[0,1].plot(screen[valleys] * 1e3, I1D[valleys], 'o', color='#781c6d', label='Minima')  

547 axes[0,1].set_title('Maxima and Minima Analysis')  

548 axes[0,1].set_xlabel('Position on Screen (mm)')  

549 axes[0,1].set_ylabel('Normalised Intensity')  

550 axes[0,1].set_ylim([-0.01, 1.05])  

551 axes[0,1].set_xlim([-1.75, 1.75])  

552  

553 axes[0,1].grid(True, which='major', linewidth=0.8, color='#DDDDDD', zorder=2)  

554 axes[0,1].grid(True, which='minor', linewidth=0.5, color='#EEEEEE', linestyle='--', zorder=1)  

555 axes[0,1].minorticks_on()  

556 axes[0,1].legend(loc='upper right')  

557  

558 # Intensity Decay Log Plot; bottom right  

559 centre = np.argmin(np.abs(screen))  

560 centre_peak = I1D[centre]  

561  

562 I1D_norm = I1D / centre_peak  

563  

564 mask = I1D_norm > 1e-12  

565  

566 ax_log = axes[1,1]  

567  

568 ax_log.semilogy(screen * 1e3, I1D_norm[mask], color='#2A2A2A', linewidth=1.5, label='Intensity  

569 Decay')  

570 ax_log.set_ylabel('Log Intensity')  

571 ax_log.tick_params(axis='y')  

572  

573 ax_lin = ax_log.twinx()  

574 ax_lin.plot(screen * 1e3, I1D, '--', color='#cb4149', linewidth=1, label='Diffraction  

575 Intensity')  

576 ax_lin.set_ylabel('Normalised Intensity')  

577 ax_lin.tick_params(axis='y')  

578  

579 ax_log.set_xlabel('Position on Screen (mm)')  

580 ax_log.set_title('Intensity Decay (Log Scale)')

```

```

574 ax_log.set_xlim([-1.75,1.75])
575
576 ax_log.grid(True, which='major', linewidth=0.8, color="#AAAAAA", zorder=2)
577 ax_log.grid(True, which='minor', linewidth=0.5, color="#EEEEEE", linestyle='--', zorder=1)
578 ax_log.minorticks_on()
579
580 lines1, labels1 = ax_log.get_legend_handles_labels()
581 lines2, labels2 = ax_log.get_legend_handles_labels()
582 ax_log.legend(lines1 + lines2, labels1 + labels2, loc='upper right')
583
584 plt.tight_layout()
585 plt.subplots_adjust(wspace=0.2, top=0.94)
586
587 plt.show()
588
589 # %% [markdown]
590 # Circular Aperture
591
592 # %%
593 # Wave Equation and Properties
594
595 R = 0.5e-3 # Radius of circular aperture
596 w = h = 2 * R
597 L = 200e-3 # Distance from aperture to screen
598 lam = 500e-9 # Wavelengths of light
599
600 k = 2 * np.pi / lam # Wave number/vector
601
602 # Defining Aperture
603 nx, ny = 1000, 1000
604 x_ap = np.linspace(-R, R, nx)
605 y_ap = np.linspace(-R, R, ny)
606 xx_ap, yy_ap = np.meshgrid(x_ap, y_ap)
607
608 dx = x_ap[1] - x_ap[0]
609 dy = y_ap[1] - y_ap[0] # Spacing between points in the aperture grid
610 area = dx * dy # Area of each point source in the aperture (Riemann sum element)
611
612 mask = xx_ap**2 + yy_ap**2 <= R**2 # Circular aperture mask
613
614 xx_ap_circ = xx_ap[mask].ravel()
615 yy_ap_circ = yy_ap[mask].ravel() # Flattening the arrays to 1D for easier computation
616
617 aperture_points = np.stack((xx_ap_circ, yy_ap_circ), axis=-1) # Creating a list of (x,y)
       points in the aperture
618
619 # Defining the Wave Equation Functions for 1D and 2D
620
621 def circ_1d(screen=(-0.75e-3, 0.75e-3), points=1000):
622
623     screen = np.linspace(screen[0], screen[1], points) # Screen
624
625     I = [] # List to hold the resultant field values at each point on the screen (intensity)
626     for x in screen:
627         r = np.sqrt((x - aperture_points[:,0])**2 + (aperture_points[:,1])**2 + L**2)
628         E = (np.exp(1j * k * r) / r) * area # Electric field at each point on the screen due
           to each point in the aperture
629         E_total = E.sum() # Total electric field
630         I.append(np.abs(E_total)**2)
631
632     I = np.array(I) # Convert list to array for easier plotting
633     I /= I.max() # Normalize intensity for plotting
634
635     return screen, I
636
637 def fraun_circ_1d(screen=(-0.75e-3, 0.75e-3), points=1000):
638
639     screen = np.linspace(screen[0], screen[1], points)
640
641     theta = screen / L # Small angle approximation
642     q = k * R * theta
643
644     with np.errstate(divide='ignore', invalid='ignore'):
645         I_fraun = ((2 * j1(q)) / q)**2
646
647     I_fraun = np.where(np.abs(q) < 1e-12, 1, I_fraun)

```

```

648     I_fraun /= I_fraun.max()
649
650     return screen, I_fraun
651
652 def fresnel_circ_1d(screen=(-0.75e-3, 0.75e-3), points=1000):
653
654     screen = np.linspace(screen[0], screen[1], points)
655
656     factor = (np.exp(1j * k * L) / (1j * lam * L)) # Fresnel Approximation
657
658     xp = aperture_points[:,0]
659     yp = aperture_points[:,1]
660
661     E_theory = np.zeros_like(screen, dtype=complex)
662     for i, x in enumerate(screen):
663         phase = np.exp((1j * k)/(2 * L) * ((x - xp)**2 + yp**2)) # Fresnel Approximation (
664             Integral Part)
665         E = phase * area
666         E_theory[i] = factor * E.sum()
667
668     I_fresnel = (np.abs(E_theory)**2)
669     I_fresnel /= I_fresnel.max()
670
671     return screen, I_fresnel
672
673 def circ_2d(screen=(-0.75e-3, 0.75e-3), screen_res=1.25e-5):
674
675     screen_x1, screen_x2 = -0.75e-3, 0.75e-3
676     screen_y1, screen_y2 = -0.75e-3, 0.75e-3
677     n = int((screen_x2 - screen_x1) / screen_res)
678     m = int((screen_y2 - screen_y1) / screen_res)
679
680     X = np.linspace(screen_x1, screen_x2, n)
681     Y = np.linspace(screen_y1, screen_y2, m)
682     X, Y = np.meshgrid(X, Y) # 2D grid of screen points
683
684     E_total = np.zeros((m, n), dtype=complex)
685     for ap in aperture_points:
686         r = np.sqrt((X - ap[0])**2 + (Y - ap[1])**2 + L**2) # ap[0] and ap[1] are the x and y
687             coordinates of the aperture point
688         E_total += (np.exp(1j * k * r) / r) * area # Sum contributions from each point source
689
690     I = np.abs(E_total)**2
691     I /= I.max()
692
693     return X, Y, I
694
695 # Patterns
696 screen, I1D = circ_1d()
697 screen, I1DF = fresnel_circ_1d()
698 screen, I1DFh = fraun_circ_1d()
699 X, Y, I2D = circ_2d(screen_res=1.25e-5)
700
701 # Plots
702 fig, axes = plt.subplots(2, 2, figsize=(14, 12), gridspec_kw={'width_ratios':[1, 1.2]})
703
704 fig.suptitle('Diffraction Pattern for a Circular Aperture (R = 0.5mm) for a 500nm Wave at 200
705 mm', fontsize=16, fontweight='bold')
706
707 # 1D Plot; top left
708 axes[0,0].plot(screen * 1e3, I1D, color="#2A2A2A", linewidth=2, label='Coded Intensity')
709 axes[0,0].plot(screen * 1e3, I1DF, '--', color="#f68d45", linewidth=1, label='Fresnel
710 Approximation')
711 axes[0,0].plot(screen * 1e3, I1DFh, '--', color="#cb4149", linewidth=1, label='Fraunhofer
712 Approximation')
713
714 axes[0,0].set_title('1D Diffraction Pattern')
715 axes[0,0].set_xlabel('Position on Screen (mm)')
716 axes[0,0].set_ylabel('Normalised Intensity')
717 axes[0,0].set_ylim([-0.01, 1.05])
718 axes[0,0].set_xlim([-0.75, 0.75])
719
720 axes[0,0].minorticks_on()
721 axes[0,0].grid(True, which='major', linewidth=0.8, color="#DDDDDD", zorder=2)
722 axes[0,0].grid(True, which='minor', linewidth=0.5, color="#EEEEEE", linestyle='--', zorder=1)
723 axes[0,0].legend()

```

```

719 # 2D Plot1; bottom left
720 graph = axes[1,0].pcolormesh(X*1e3, Y*1e3, I2D, shading='auto', cmap='inferno')
721
722 axes[1,0].set_aspect('equal')
723 axes[1,0].set_title('2D Diffraction Pattern')
724 axes[1,0].set_xlabel('x-axis of screen (mm)')
725 axes[1,0].set_ylabel('y-axis of screen (mm)')
726
727 fig.colorbar(graph, ax=axes[1,0], orientation='vertical', label='Normalised Intensity', shrink
    =0.8)
728 axes[1,0].minorticks_on()
729
730 # Peak and Valley Slices; top right
731 # https://docs.scipy.org/doc/scipy/reference/generated/scipy.signal.find_peaks.html#scipy.
    signal.find_peaks
732 peaks, _ = find_peaks(I1D, height=0.01)
733 valleys, _ = find_peaks(-I1D, prominence=0.005)
734
735 axes[0,1].plot(screen * 1e3, I1D, color='#2A2A2A', linewidth=1.5)
736 axes[0,1].plot(screen[peaks] * 1e3, I1D[peaks], 'o', color='#fca50a', label='Maxima')
737 axes[0,1].plot(screen[valleys] * 1e3, I1D[valleys], 'o', color='#781c6d', label='Minima')
738 axes[0,1].set_title('Maxima and Minima Analysis')
739 axes[0,1].set_xlabel('Position on Screen (mm)')
740 axes[0,1].set_ylabel('Normalised Intensity')
741 axes[0,1].set_ylim([0, 1.05])
742 axes[0,1].set_xlim([-0.75, 0.75])
743
744 axes[0,1].grid(True, which='major', linewidth=0.8, color="#DDDDDD", zorder=2)
745 axes[0,1].grid(True, which='minor', linewidth=0.5, color="#EEEEEE", linestyle='--', zorder=1)
746 axes[0,1].minorticks_on()
747 axes[0,1].legend(loc='upper right')
748
749 # Intensity Decay Log Plot; bottom right
750 centre = np.argmin(np.abs(screen))
751 centre_peak = I1D[centre]
752
753 I1D_norm = I1D / centre_peak
754
755 mask = I1D_norm > 1e-12
756
757 ax_log = axes[1,1]
758
759 ax_log.semilogy(screen * 1e3, I1D_norm[mask], color='#2A2A2A', linewidth=1.5, label='Intensity
    Decay')
760 ax_log.set_ylabel('Log Intensity')
761 ax_log.tick_params(axis='y')
762
763 ax_lin = ax_log.twinx()
764 ax_lin.plot(screen * 1e3, I1D, '--', color='#cb4149', linewidth=1, label='Diffraction
    Intensity')
765 ax_lin.set_ylabel('Normalised Intensity')
766 ax_lin.tick_params(axis='y')
767
768 ax_log.set_xlabel('Position on Screen (mm)')
769 ax_log.set_title('Intensity Decay (Log Scale)')
770 ax_log.set_xlim([-0.75, 0.75])
771
772 ax_log.grid(True, which='major', linewidth=0.8, color="#DDDDDD", zorder=2)
773 ax_log.grid(True, which='minor', linewidth=0.5, color="#EEEEEE", linestyle='--', zorder=1)
774 ax_log.minorticks_on()
775
776 lines1, labels1 = ax_log.get_legend_handles_labels()
777 lines2, labels2 = ax_lin.get_legend_handles_labels()
778 ax_log.legend(lines1 + lines2, labels1 + labels2, loc='upper right')
779
780 plt.tight_layout()
781 plt.subplots_adjust(wspace=0.2, top=0.94)
782
783 plt.show()

```