behaviour. Firstly, you can set a desired accuracy for the solution: setting a higher accuracy may make the program run more slowly because it will typically take shorter steps. Secondly, the functions usually have the capability to return an interpolated value of the solution at user-specified times, and not just at the time steps used to integrate the ODE. You can choose a time sampling to give appropriately smooth plots. It pays to experiment with these values to see if they have any effect on your results, especially when investigating "chaotic" behaviour.

6. To find the period versus amplitude relationship, a simple (and just about acceptable) way is to measure the period off a suitable plot, and do this for several values of θ_0. However, it is much better to alter your code to estimate the period directly. You can then loop over θ_0 values and plot the period versus amplitude relation. Two obvious approaches spring to mind. First, you can find when θ first goes negative. This is when the time is approximately $T/4$ where T is the period. How accurate would this result be? You could also find several zero crossings by considering when y changes sign or becomes exactly zero after a step is taken; by counting many such zero crossings and recording the time between them you can get a more accurate value for T.
7. Note that you need to think about what happens when the pendulum goes "over the top" and comes down the other side — you need to think about the 2π ambiguities involved, and what this means in terms of the "period" of an oscillation.

Exercise 2: Fraunhofer and Fresnel Diffraction

Goal

Write a program to calculate the near and far-field diffraction patterns of an arbitrary one-dimensional complex aperture using the Fast Fourier Transform technique. Test this program by using simple test apertures (a slit) for which the theoretical pattern is known. Investigate more complicated apertures for which analytical results are difficult to compute.

Physics

Plane monochromatic waves, of wavelength λ, arrive at normal incidence on an aperture, which has a complex transmittance $A(x)$. The wave is diffracted, and the pattern is observed on a screen a distance D from the aperture and parallel to it. We want to calculate the pattern when the screen is in the far-field of the aperture (Fraunhofer diffraction) and also in the near-field (Fresnel).

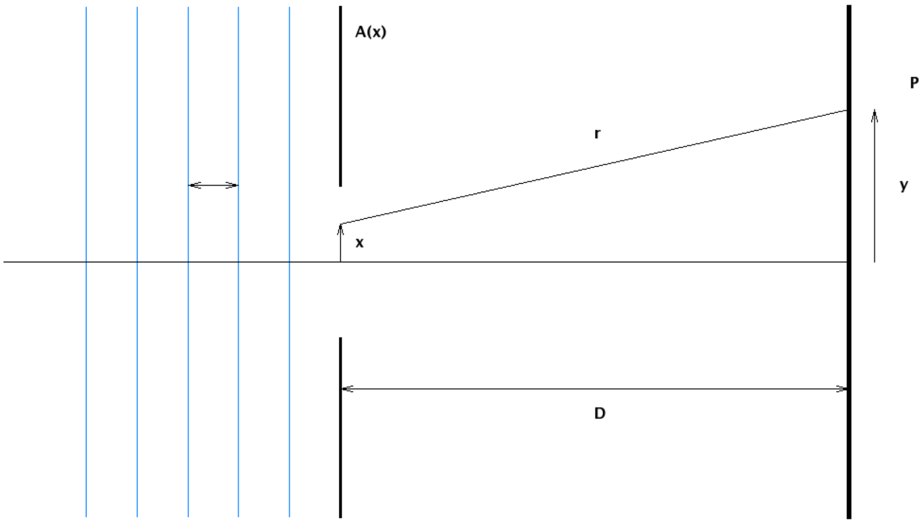

Figure 1: Geometry for diffraction calculation

Using Huygen's construction, we can write the disturbance at a point P on the screen, a distance y from the axis, as

$$\psi(y) \propto \int \frac{A(x)\, \exp(ikr)}{r}\, \mathrm{d}x$$

where $k = 2\pi/\lambda$. We have assumed that all angles are small:

$$x, y \ll D$$

so that we are close to the straight-through axis and can therefore neglect terms like $\cos(\theta)$ which appear if we are off-axis. We now expand the path length r in powers of x/r:

$$r^2 = D^2 + (y - x)^2$$

$$r \approx D + \frac{y^2}{2D} - \frac{xy}{D} + \frac{x^2}{2D} + \mathcal{O}\left(\frac{(y-x)^4}{D^3}\right)$$

If we now neglect the variation in r in the denominator of the integral, setting $r \approx D$, which is adequate for $x, y \ll D$, then we can write

$$\psi(y) \propto \frac{\exp(ikD)}{D}\, \exp\left(\frac{iky^2}{2D}\right) \int A(x)\, \exp\left(\frac{ikx^2}{2D}\right) \exp\left(\frac{-ikxy}{D}\right) \mathrm{d}x \tag{3}$$

The diffraction pattern is thus the Fourier-transform of the modified aperture function A':

$$\psi(y) \propto \exp\left(\frac{iky^2}{2D}\right) \int A'(x)\ \exp\left(\frac{-ikxy}{D}\right) \mathrm{d}x \tag{4}$$

with

$$A'(x) = \exp\left(\frac{ikx^2}{2D}\right) A(x) \tag{5}$$

In the far-field (Franhofer limit) we have $kx^2/(2D) \ll \pi$ so that $A' \approx A$ for all values of x in the aperture where $A(x)$ is non-zero, i.e. the familiar result

$$d \gg \frac{x^2_{\text{max}}}{\lambda}.$$

The distance x^2_{max}/λ is the Fresnel distance. In this case, the diffraction pattern is just the Fourier transform of the aperture function.

Note that we can calculate the near-field (Fresnel) pattern also if we include a step to modify the aperture function according to Equation 5 *before we take its Fourier transform*.

Note that if we are only interested in the pattern's intensity, we can ignore the phase prefactor in Equation 4.

Finally, we can discretize Equation 4, by sampling the aperture evenly at positions x_j

$$\psi(y) \propto \Delta \sum_{j=0}^{N-1} A'(x_j)\ \exp\left(\frac{-ikx_j y}{D}\right) \tag{6}$$

where Δ is the distance between the aperture sample positions x_j.

One convenient definition of the sample points in x is

$$x_j = (j - (N/2))\Delta, \tag{7}$$

where N is the number of sample points in the aperture. Note that this definition of the x-coordinate is equivalent to applying a coordinate transform equivalent to the "fftshift" operation described in the lectures, and results in Fourier phases which are closer to zero compared to a more simple linear relationship between x_j and j.

Tasks

Core Task 1 : Write a program that will calculate the diffraction pattern of a general 1-dimensional complex aperture in the far field of the aperture using FFT techniques. The program should calculate the intensity in the pattern across the screen, which you should plot using the correct y coordinates (in metres or microns for example). **Label your coordinates.**

Test this program for the specific case of a slit in the centre of an otherwise blocked aperture: take the single slit to have width d in the centre of an aperture of total extent L. For definiteness, use $\lambda = 500$ nm, $d = 100\,\mu$m, $D = 1.0$ m and $L = 5$ mm. Overlay on your plot the theoretical value of the intensity pattern expected.

Core Task 2 : Now calculate and plot the Fraunhofer diffraction pattern of a *sinusoidal phase grating*. This grating is a slit of extent $d = 2$mm, outside of which the transmission is zero. Within $|x| < d/2$, the transmission amplitude is 1.0, and the phase of A is

$$\phi(x) = (m/2)\sin(2\pi x/s)$$

where s is the spacing of the phase maxima, and can be taken as 100microns for this problem. For this calculation, use $m = 8$. The Fresnel distance d^2/λ is 8 m, so calculate the pattern on a screen at $D = 10$ m. What do you notice about the resulting pattern?

Core task 3 : Now modify your program so that the calculation is accurate even in the near-field by adding a phase correction to the aperture function as defined by Equation 5. Repeat your calculations in the previous two tasks for $D = 5$ mm for the slit, and $D = 0.5$ m for the phase grating, and plot the results. Do the intensity patterns look sensible?

Supplementary Task 1 : Write a program to evaluate the Fresnel integrals *accurately* using a standard integration routine from the `scipy`, and use this to make a plot of the Cornu spiral using `pyplot`. Do not use a Monte-Carlo routine — they are not efficient for low-dimensional integrals — instead use a standard quadrature technique. One version of the Fresnel integrals can be written

$$C(u) = \int_0^u \cos\left(\frac{\pi\,x^2}{2}\right)\,dx$$

$$S(u) = \int_0^u \sin\left(\frac{\pi\,x^2}{2}\right)\,dx$$

Note that there is a function `scipy.special.fresnel()` whose only purpose is to evaluate this integral! However using this special-purpose function to compute your answer negates the point of understanding how to use a general-purpose integrator and will not be looked on favourably.

Supplementary Task 2 : Use the Fresnel integrals computed in Supplementary Task 1 to re-calculate the diffraction pattern of the near field slit in Core Task 3 (not the phase grating). Physics reminder: the complex amplitude in the near field on the axis of an open slit is given by

$$\Psi \propto \int_{x_0}^{x1} \cos(\pi x^2/(\lambda D)) + i \sin(\pi x^2/(\lambda D))\, dx,$$

so you can use the Fresnel integrals directly by scaling the length dimensions by $\sqrt{2/(\lambda D)}$. To calculate the intensity pattern as we move off-axis, we can imagine translating the slit relative to the screen and change the integration limits appropriately. Compare the results for this task with the results from the Fourier method in Core Task 3.

Hints

Diffraction calculations using the FFT Recall the DFT definition:

$$H_j = \sum_{m=0}^{N-1} h_m \, e^{2\pi \, i \, m \, j/N} \tag{8}$$

which maps N time-domain samples h_m into N frequencies, which are

$$f_j = \frac{j}{N\Delta} \tag{9}$$

You can think of frequencies $(j/N) \times (1/\Delta)$, running from $j = 0$ to $(N-1)$, with

- $j = 0$ is zero frequency.
- For $1 \leq n \leq (N/2)$, we have positive frequencies $(j/N) \times (1/\Delta)$.
- For $(N/2) + 1 \leq j \leq (N-1)$ we have *negative frequencies* which we compute as $((j/N) - 1) \times (1/\Delta)$. (Remember the sequences are periodic).

Of course in this case we don't have time-domain samples, but we can still use the FFT to carry out the transform.

The complex aperture function will be represented by an array of N discrete complex values along the aperture, encoding the real and imaginary parts of $A(x)$. Each complex value represents the aperture's transmittance over a small length Δ of the aperture, so that $N\Delta$ is the total extent of the aperture.

Choose appropriate values for N and Δ to make sure you can represent the whole aperture of maximum extent L well enough. Bear in mind that Fast Fourier Transform calculations are fastest when the transform length is a power of 2, and that you want Δ to be small enough to resolve the features of the aperture. (Computers are fast these days though; so you can use small values of Δ and correspondingly large values of N. In practice, for such a small problem, the use of N as a power of 2 is not necessary, but it is important if performance is critical.)

For the slit problem you can use the `numpy.zeros()` function to set up an array of the appropriate size filled with zeros and then set the locations where the slit is transparent by assigning non-zero values to "slices" e.g. `a[5:10]=1.0`.

You now need a routine to calculate the fast Fourier transform (FFT). You can use the `numpy.fft.fft()` function, which is straightforward to use (see the code examples from the lectures). It accepts a real or a complex input and produces a complex output. You will need to compute the intensity of the pattern. You might also want to plot the aperture function to make sure you have calculated it correctly.

Now think carefully about the coordinates associated with your calculated pattern. The discussion at the beginning of this section reminds you about how the frequencies appear in the FFT'ed data. Can you understand the form of the intensity pattern you have derived?

To plot the intensity on the screen as a function of actual distance y, you need to work out how to convert the pixels in the Fourier transform into distances on the screen y. To do this you first need to compare carefully Equation 6 and Equation 8 (also referring to Equation 7) which should tell you how to derive y at each pixel value. In addition, by interpreting the second half of the transform as negative frequencies (or y values in this case) you should be able to plot the intensity pattern as a function of y for positive and negative y, and plot over this the matching sinc function for a slit. **If you are having difficulties with this step it may help to revise your notes from the lectures concerning the location of negative frequencies in the output of an FFT**.

Numerical Integration with `scipy` You will need to choose a `scipy` routine to do the integration. The `scipy.integrate.quad()` function is a good general purpose integrator. Example code is available online: search for "scipy numerical integration examples".

Write suitable functions that evaluate the two integrands. Evaluate the integrals for various values of s and use `pyplot` to plot the spiral.

Plotting the diffraction pattern should now be straightforward. Remember to think carefully about the coordinates associated with your calculated pattern.