Diffraction and Fourier Transforms Phys 423

In class we have seen the relationship between diffractions patterns and Fourier transforms. For this lab we will use the discrete Fourier transform to calculate the diffraction pattern in three situations. The first two are one-dimensional and are the double slit and the single slit with a phase element that we examined in the second homework. The third situation is the two-dimensional rectangular slit.

A quick word on terminology: the discrete Fourier transform happens when you digitally calculate the Fourier transform, the Fast Fourier transform (FFT) is an algorithm that quickly implements a discrete Fourier transform.

You should review the “help’ in MATLAB on the FFT.

You might remember that in Physics 323 we simulated the diffraction from a slit by following the waves from Huygenian emitters from the aperture out to the observation screen. Why can we not simply continue with this, and not bother about discrete Fourier transforms? One good reason to go with the digital Fourier transform is speed. Imagine you want to propagate light from N points on a line to N points on another. To do this using the direct method would take N^2 mathematical operations. But it can be shown that using the FFT the number of operations is about $N\log_2 N$, which can represent a huge savings in time. Going beyond optics, the discrete Fourier transform is ubiquitous in signal processing and is also used in numerical solution of differential equations.

Introduction to Fourier transforms and power spectra

In this course and others, you have used the Fourier transform to convert a function from, say, the time domain to the frequency domain. The Fourier transform of the temporal signal I(t) can be defined as

$$F(f) = \int_{-\infty}^{\infty} I(t) e^{-2\pi i f t} dt$$

where we are using frequency rather than angular frequency. This complex valued function of the frequency contains the information on the amplitudes and phases of the harmonics that make up the original signal. Often however, we are just interested in the “strength” of the harmonics which we get by multiplying the Fourier transform by its own complex conjugate so that the power spectrum is given by

$$P(f) = \left|F(f)\right|^2$$

In the analysis of signals, we might not have an expression for I(t) or even if we do then we cannot do the integral so it becomes necessary to numerically calculate an estimate for the power spectrum. You have perhaps done something like this in math classes where you take the signal, multiply it by various sines and cosines and then compute the integral for each of the sines and cosines. This will work but it is a very time-consuming process. Fortunately, in 1965, Cooley and Tukey invented the Fast Fourier Transform (FFT), an algorithm which computes the Fourier transform and does it much faster than the direct approach. This algorithm is not easy to program but, again fortunately, it is available as a subroutine in many different computer languages. All you need to do is supply the input signal and read out the transform. MATLAB has the FFT built in. The following code snippet generates a sine wave and computes and plots the power spectrum. I have put in numbers to correspond to seconds and hertz so you can visualize what might happen with real signals.

```
%using the FFT to calculate a power spectrum
%This program generates a sine wave and you can adjust the frequency
%by changing the variable "freq".
%It then computes the power spectrum and the program shows
%you how to figure the correct scaling for the frequency axis.
%jps 2021

%first, make a signal
num_pts=2048; %the signal will be num_pts long
tmax=1; %the time will run from 0 to tmax seconds
t=linspace(0, tmax, num_pts); %this is the time axis
freq=50; %frequency, hertz. See what happens when you change this
s=sin(2*pi*freq*t);  % this is the signal
subplot(2,1,1)
plot(t,s);
xlabel('time (s)'); ylabel('signal (arb)')

%this next bit is to figure the frequency scale for the Fourier transform
%the fundamental frequency is when one whole sine "fits" the
%total length of the signal.
fund=1/tmax;
f=fund*(0:num_pts-1); %this is the frequency axis

spect=fft(s); %this is the spectrum - it is a complex array

ps=spect.*conj(spect); %this is the power spectrum
subplot(2,1,2)
plot(f(1:num_pts/2),ps(1:num_pts/2)); %We only plot the first half of
%the spectrum. It repeats itself.
xlabel('frequency (Hz)'); ylabel('Intensity (arb)');
```

Double slit

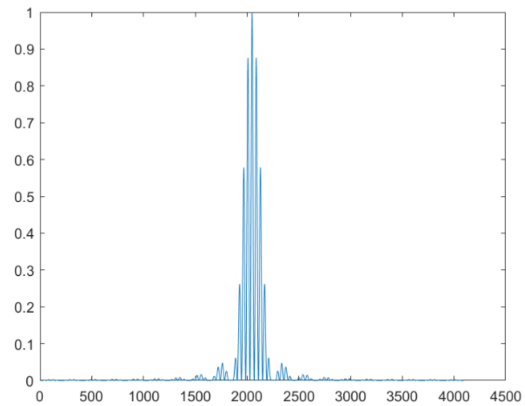
Now that we have a feeling for the Fourier transform, we can turn to the double slit illuminated with a plane wave. We need to generate a representation of the transmission of the double slit aperture. This could be done something like:

```
sze=4096;
slts=zeros(1,sze);

s_sep=100;
s_wid=20;

slts(sze/2-s_sep/2-s_wid/2+1:sze/2-s_sep/2+s_wid/2)=ones(1,s_wid);
slts(sze/2+s_sep/2-s_wid/2+1:sze/2+s_sep/2+s_wid/2)=ones(1,s_wid);
```

Then this is transformed and the power spectrum calculated. However, you will notice that the power spectrum looks a little odd. This is because the raw output of the FFT algorithm is arranged in a non-obvious way. What you need to do is apply the command fftshift (check the help in MATLAB) to rearrange the data. So, taking the FFT and computing the power spectrum would be

```
f=fft(slts); f=fftshift(f);
I=f.*conj(f); I=I/max(I);
```

Plotting the irradiance will give the following:



Which should look pretty familiar. What you will need to do is figure out what horizontal axis should go on this graph. Recall from the discussion of temporal Fourier transforms that the first frequency is the reciprocal of the length of the signal in time. Our spatial frequencies will depend on the size of the domain that the slits are embedded in. To make things concrete. let's imagine that the slits are separated by 100 microns and have a width of 20 microns, the light wavelength is 500 nm and the screen on which we observe the diffraction pattern is 2 m from the slits. The fundamental frequency returned by the FFT will correspond to 1/(length of the space domain). Note that there is a significant difference between the transform that we calculate for temporal signals and for spatial signals. For the spatial case we have both positive and negative frequencies – which correspond to the positive and negative orders of diffraction. The zero of frequency thus corresponds to the middle of the irradiance graph shown above.

Once you have computed the diffraction pattern using the FFT you should look up the analytical formula in the textbook and use it to calculate the diffraction pattern as well. The two results could be plotted on the same graph and should be pretty similar.

Single slit with a phase element

Now we are going to compute the diffraction pattern from an object that modulates the phase of the input light. We will use the structure in question 2 in Homework 2 where there is a phase modulating element in the middle of a single slit. We found there that the irradiance as a function of angle was proportional to

$$\left[\frac{\sin(\frac{kd}{2}\sin(\theta))}{(\frac{kd}{2}\sin(\theta))}-\frac{\sin(\frac{kd}{4}\sin(\theta))}{(\frac{kd}{4}\sin(\theta))}\right]^2$$

Where the overall slit width is d and the width of the phase-shifting element was d/2.
Recall how we represent the presence of the phase element. It introduces a 180° phase shift and this can be written as a multiplication by -1. Again, to make the problem concrete make the slit 50 microns wide and the phase element 25 microns wide, with light wavelength 500 nm and screen at 1 m distant.

Two-Dimensional diffraction

In this case we will compute the diffraction pattern due to a rectangular slot cut in an otherwise opaque screen. Perhaps you can start with something like this for the aperture.

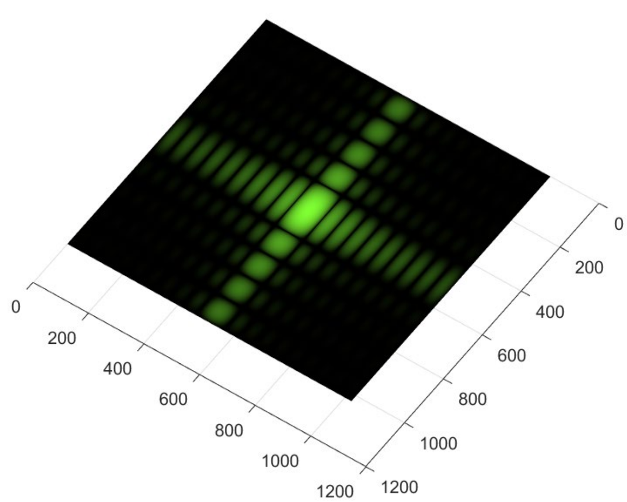
```
sze=1024;
slot_sze_x=20;
slot_sze_y=10;

aper=zeros(sze,sze);

slot=ones(slot_sze_x,slot_sze_y);

aper(sze/2-slot_sze_x/2:sze/2+slot_sze_x/2-1,sze/2-
slot_sze_y/2:sze/2+slot_sze_y/2-1)...
    =slot;

imagesc(aper);
```

While you could use one dimensional FFT's operating first on the rows and then the columns of the object to be transformed, MATLAB provides a two-dimensional FFT, called fft2. Your output data will be a two-dimensional array.

A way to display the data is a surf plot. If you use the option interp for shading that will give a nice picture, and you could get something like the image below with the command line

```
surf(lp); shading interp; view(125,90)
```

where lp is a two-D array. Note how the pattern is stretched out in one direction and compressed in the other. This is because the slot is wider in one direction than the other.



You will have to use a "colormap" to map your values of intensities into colors. A good one is "gray" though it is pretty easy to make your own. The one above is

```
gr=zeros(256,3);
gr(:,1)=(0:255)/(255*2);
```

```
gr(:,2)=(0:255)/(255*1.0);
gr(:,3)=(0:255)/(255*5);
colormap(gr)
```

Report

Your report should explain the connection between the diffraction integral that we can obtain from appropriate simplification of the Kirchoff diffraction integral, and how this integral can be interpreted as a Fourier transform of the object transmission function when the object is illuminated with a uniform plane wave. Then the report should include:

- Code for 1-dimensional double slits
- Graph of the irradiance pattern one would get where the slits are separated by 100 microns and have a width of 20 microns, the light wavelength is 500 nm and the screen on which we observe the diffraction pattern is 2 m from the slits. The graph should be superimposed on the graph obtained from a plot of the analytical expression for the irradiance due to this double slit geometry.
- Code for 1-dimensional single slit with phase element (see Q2 in homework 2) where the slit width is 50 microns, the phase element is 25 microns wide, the wavelength is 500 nm, and the observing screen is 1 m distant.
- Graph of the irradiance pattern, superimposed on a plot of the irradiance using the analytical expression obtained in Homework 2.
- Code for 2-dimensional aperture
- Image of the diffraction pattern due to a slot of 20 microns x 10 microns, with a wavelength of 500 nm and a screen 1 m distant.