Daniel Guzi, Isabelle Leonard, Xinping Zhang

Optics 211

4/29/20

Final Project

**Introduction**

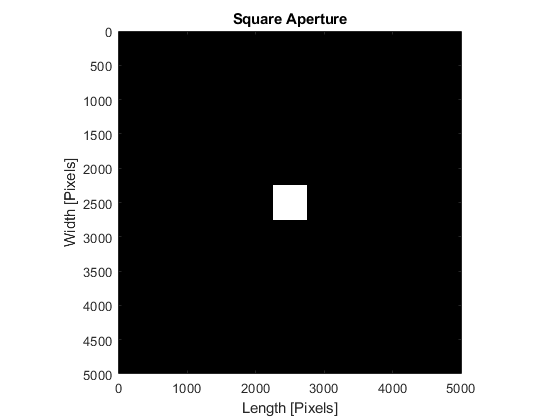
The goal of this project was to simulate far-field diffraction of different apertures. To do so, we had to generate a script that would create the aperture and then apply the Fourier transform to it create a model of the far-field diffraction pattern. Having a simulation of far-field diffraction serves many purposes. For example, with a simulation, it may be easier to see a clearer image of the diffraction pattern because you can scale down the intensity of the diffraction, allowing you to see the less prominent features of the diffraction. Another purpose is to help demonstrate what causes far-field diffraction patterns to be what they are, without having to set up and continuously change a practical set up. This would also allow people who do not have the resources to preform the actual experiment to see what different diffraction patterns look like.

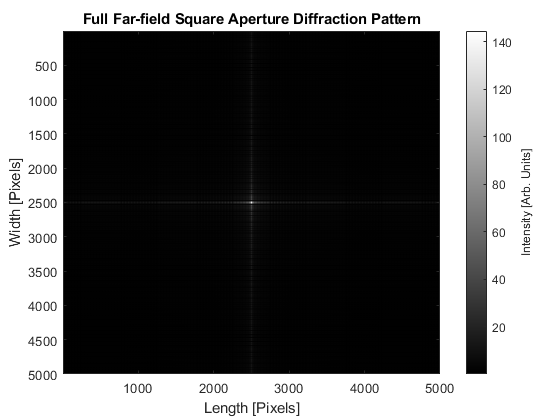
**Script Description**

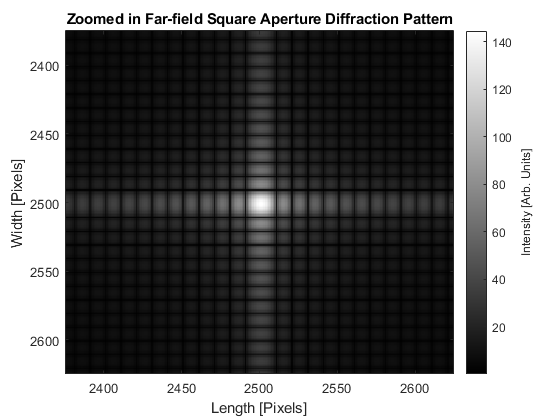
* The first line allows for a user input
* Then there is the header and help file
* Next the aperture field is generated (which will be modified later)
  + This is a 5000x5000 zeros matrix, where each value represents a pixel
* If/elseif statements
  + The purpose of these statements are to determine what aperture the user inputted
  + Within each statement, there is the code to define the aperture itself by modifying the initial aperture field
    - This is done through for loops
      * The for loops go through and check each pixel to see if it is within the desired area for the aperture and changes the value of to 1 if it is
  + The else statement at the end is in case the user inputs an undefined aperture and generates an error statement that gives the acceptable apertures
* The next section plots the aperture field as an image and changes various properties of the plot
* The next section calculates the diffraction pattern
  + The Fourier transform is taken, then the real part is taken
  + The intensity is than scaled so that the change in intensity is small enough to be effectively modeled by the grayscale colormap in MATLAB
* The next section plots the full-far-field diffraction as an image and defines the properties of the plot
* The next section plots a close up of the center of the diffraction pattern and defines the properties of the plot
* The next two sections plot the horizontal and vertical intensity cross sections in their own figure windows
  + The properties of the respective plots are also defined in these sections

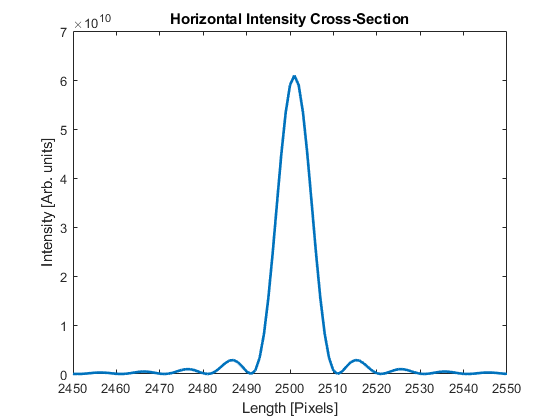
**Results**

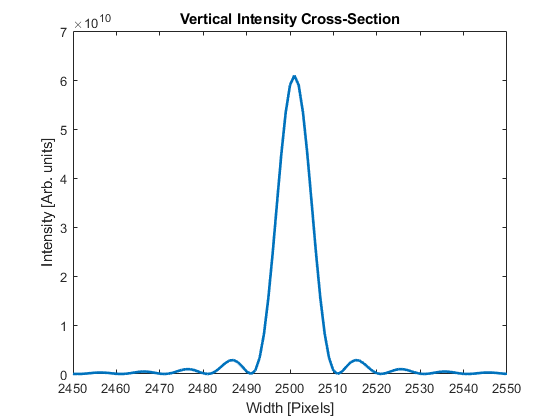
*Square Aperture:*



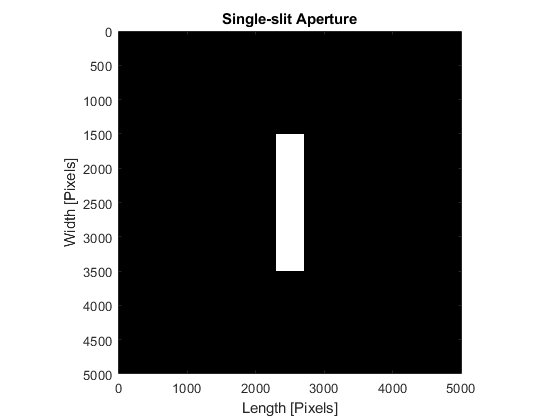


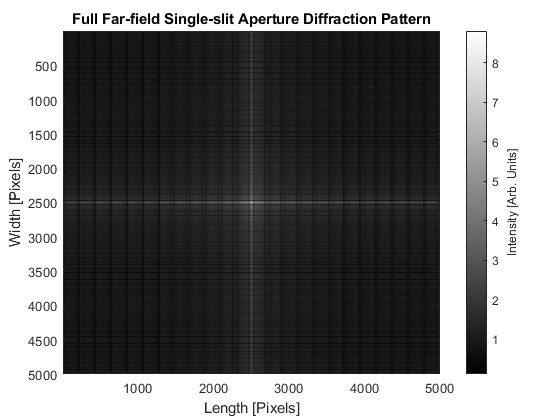


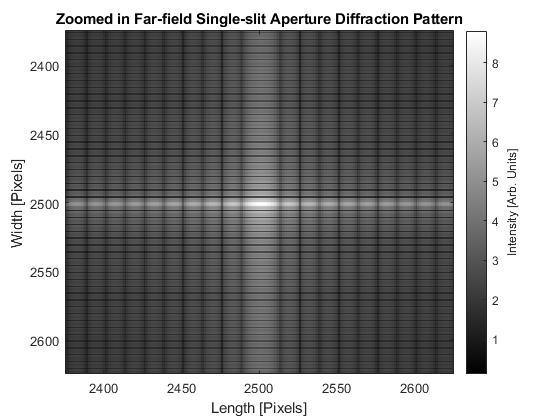


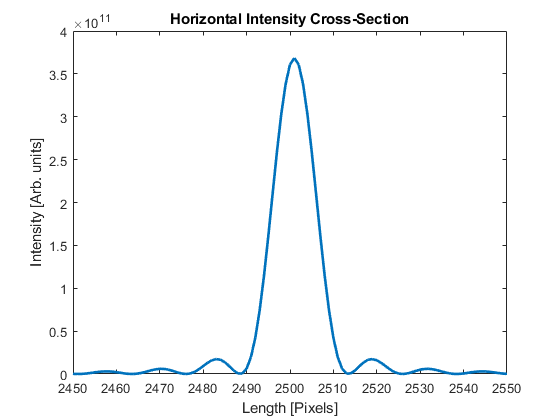


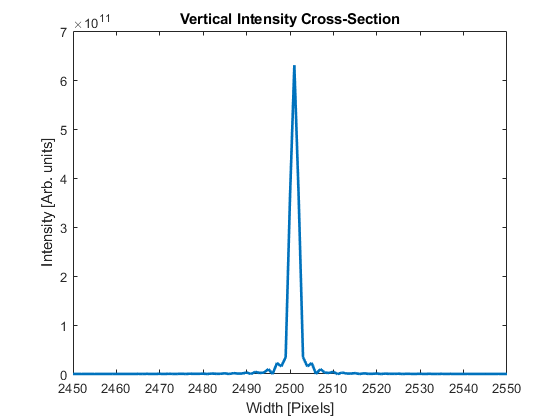
*Single-slit Aperture:*



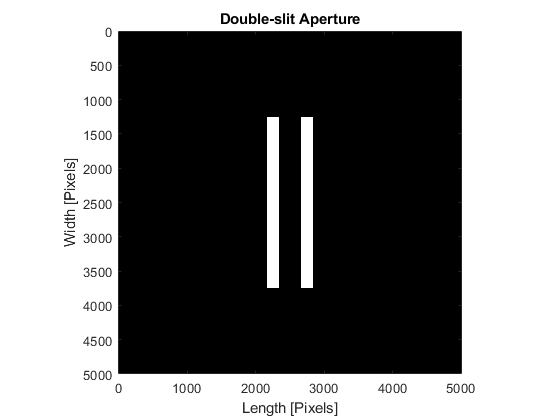


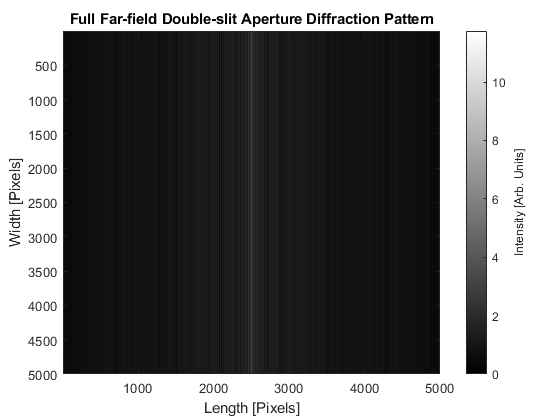


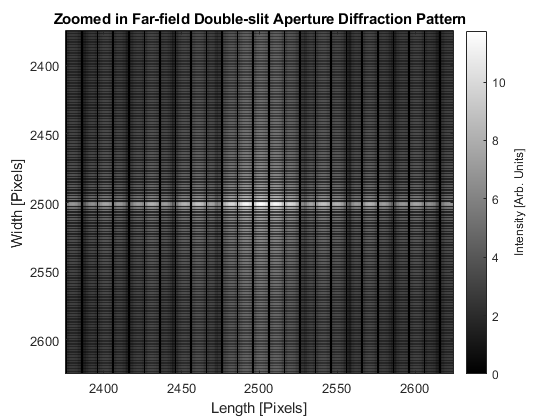


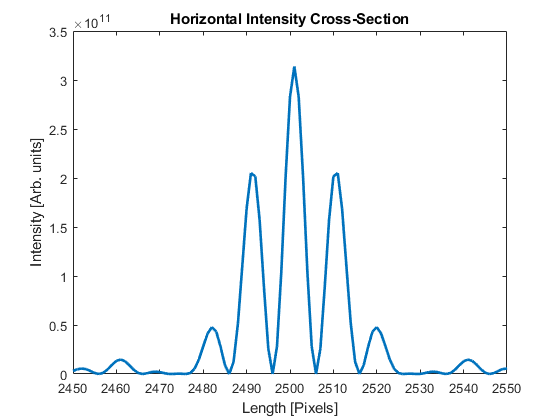


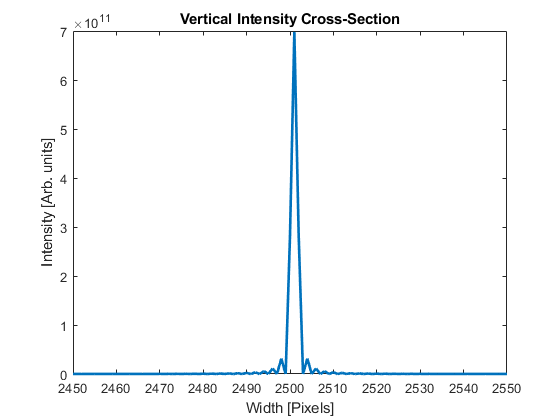
*Double-slit Aperture:*



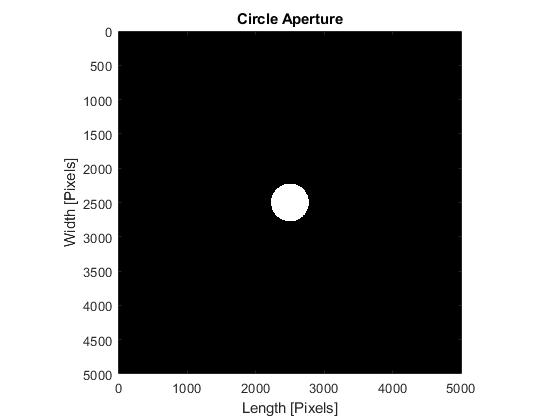


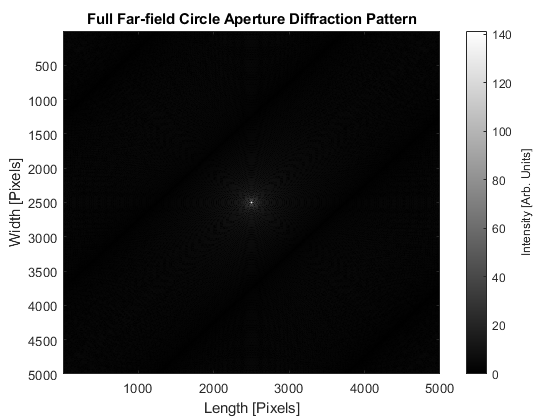


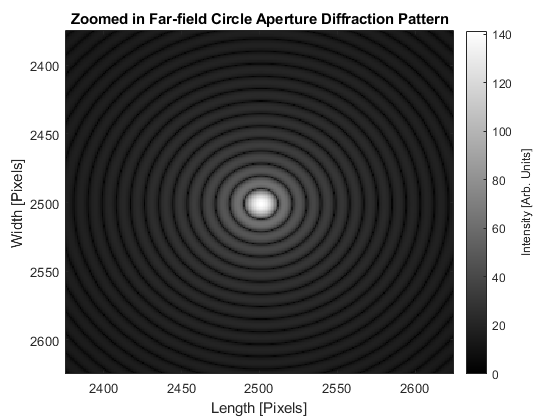


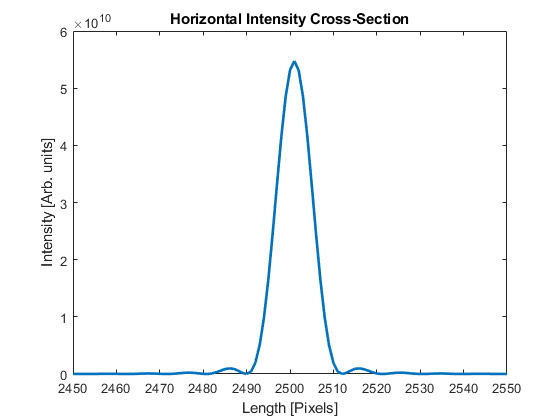


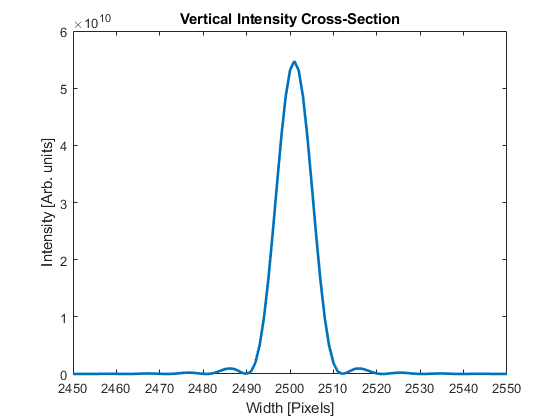
*Circle Aperture:*



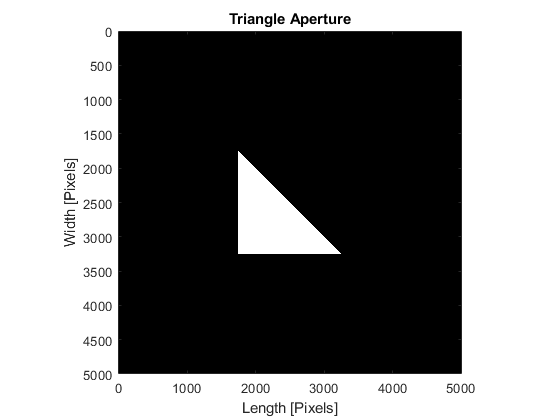


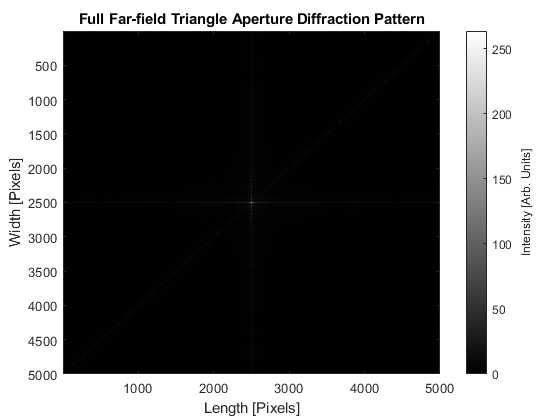


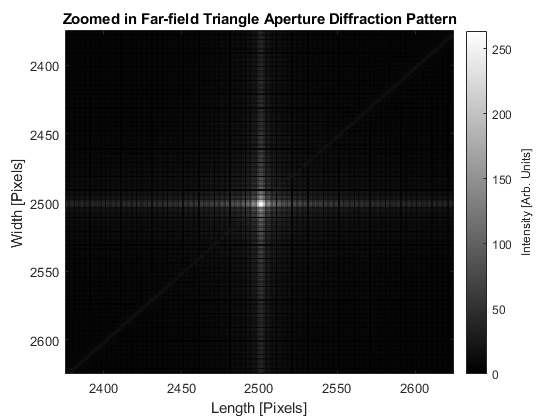


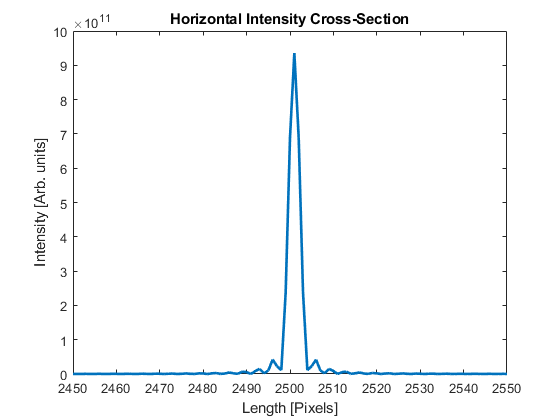


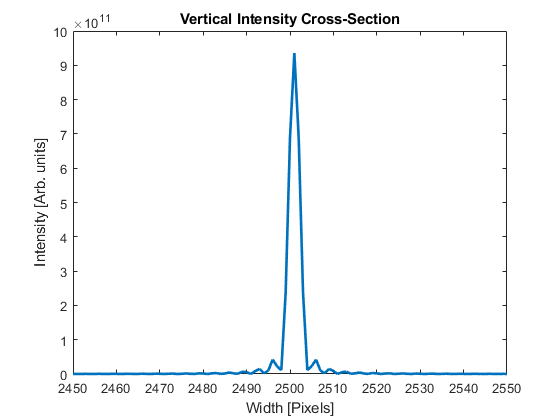
*Triangle Aperture:*











**Discussion and Conclusions**

The results of the simulation tell us that far-field diffraction is fairly predictable. The results tell us this because of the shared traits among the different apertures. All the apertures show us that the far-field diffraction pattern always perpendicular to the edges of the aperture. For example, in the triangle aperture, you get a resemblance of the square aperture’s diffracting pattern, but you also get a faint line that is perpendicular to the sloped edge of the aperture. While you can predict the general picture of what a diffraction pattern will be, you cannot, without previous examples, predict where any interference occurs, and therefore more specific intensities. Interference plays an interesting part in the specifics of the pattern as its what causes some areas to have a small or no intensity (destructive interference) while other areas have a much higher intensity. What we found was that apertures that had more parallel edges, the larger the difference in intensities between the extremes of interference. A good example of this is the difference in the intensities between the square and triangle apertures. While the two apertures had similar diffraction patterns, the maximum intensity of the triangle’s diffraction was not as great as the squares, because the triangle did not have parallel sides.

**Appendix 1**

function FFDiffraction(apt)

%Daniel Guzi, Isabelle Leonard, Xinping Zhang

%Optics 211

%4/27/20

%Final Project

%

%

% This script will generate certain apertures, their far field diffraction

% patterns, and certain cross sections of their intesities.

%

% This script is a function of one user input, and should be run as the

% following:

%

% FFDiffraction(apt)

%

% Where apt the aperture desired (one of five options) and should be

% entered as follows:

%

% 'Square'

% 'Single-slit'

% 'Double-slit'

% 'Circle'

% 'Triangle'

%

% When the script is run, it will generate five plots:

% -The aperture

% -The full far field diffraction pattern

% -A zoomed in view of the diffraction pattern

% -A horizontal intensity cross section

% -A vertical intensity cross section

% or an error message will be displayed if the input is incorrectly entered.

% Define Aperture field

apl=5000; % size of the aperture field

ap=zeros(apl); % Define actual aperture plane

%Generate correct aperture from user input

if strcmp(apt,'Square')==1

sql=500; % Define square length

for ijk=round(1+apl/2-sql/2):round(1+apl/2+sql/2)

for lmn=round(1+apl/2-sql/2):round(1+apl/2+sql/2)

ap(ijk,lmn)=1;

end

end

e = 0.2; %For scaling diffraction

elseif strcmp(apt,'Single-slit')==1

% Define single slit aperture

sql=2000; % Define slit length

for ijk=round(1+apl/2-sql/2):round(1+apl/2+sql/2)

for lmn=round(1+apl/2-sql/10):round(1+apl/2+sql/10)

ap(ijk,lmn)=1;

end

end

e = 0.08;

elseif strcmp(apt,'Double-slit')==1

% Define double slit aperture

sh=2500; % Define slit height

sw=175; % Define slit height

space=500; % Define space between the centers of the slits

for ijk=round(1+apl/2-sh/2):round(1+apl/2+sh/2)

for lmn=round(1+apl/2-space/2-sw/2):round(1+apl/2-space/2+sw/2);

ap(ijk,lmn)=1;

end

for lmn=round(1+apl/2+space/2-sw/2):round(1+apl/2+space/2+sw/2);

ap(ijk,lmn)=1;

end

end

e = 0.09;

elseif strcmp(apt,'Circle')==1

% Define circle aperture

radius=275; % Define radius of circle

for ijk=1:apl;

for lmn=1:apl;

if (lmn-apl/2).^2+(ijk-apl/2).^2<=radius.^2;

ap(ijk,lmn)=1;

end

end

end

e = 0.2;

elseif strcmp(apt,'Triangle')

% Define triangle aperture

sl=1500; % Define the side length of the triangle

for ijk=round(1+apl/2-sl/2):round(1+apl/2+sl/2)

for lmn=round(1+apl/2-sl/2):ijk;

ap(ijk,lmn)=1;

end

end

e = 0.2;

else

disp('Error: Please enter one of the allowed commands as specified by below.')

disp('''Square'', ''Single-slit'', ''Double-slit'', ''Circle'', or ''Triangle''')

return

end

%plot aperture field

figure

imagesc(ap) % Plot image of the aperture field

colormap gray % Set the color of the aperture field plot

axis equal % Set the display scale of the axes

axis([0 apl 0 apl]) % Set axes limits to size of aperture field

xlabel('Length [Pixels]')

ylabel('Width [Pixels]')

title(apt + " Aperture")

%Diffraction Pattern

f1=(real(fft2(ap))); % The real component of the Fourier Transform

f2=fftshift(f1); % Shift the zero frequency to the center

I1=f2.^2; %square the real part of fourier transform to get the Intensity

I2=I1.^(e); %the scaled intensity

%plot the diffraction pattern

figure(2)

imagesc(I2) %plot image of the full far field diffraction pattern

colormap gray % Set the color of the diffraction plot

xlabel('Length [Pixels]')

ylabel('Width [Pixels]')

title("Full Far-field " + apt + " Aperture Diffraction Pattern")

c1 = colorbar; %Show colorbar

c1.Label.String = 'Intensity [Arb. Units]'; %Label colorbar

%plot zoomed in diffraction pattern

figure(3)

imagesc(I2) %plot image of the full far field diffraction pattern

colormap gray % Set the color of the diffraction plot

xlabel('Length [Pixels]')

ylabel('Width [Pixels]')

title("Zoomed in Far-field " + apt + " Aperture Diffraction Pattern")

c2 = colorbar; %Show colorbar

c2.Label.String = 'Intensity [Arb. Units]'; %Label colorbar

xlim([2375,2625])

ylim([2375,2625])

% Plot the horizontal cross-section

figure(4)

x=linspace(1,5000,5000);

y=I1(2500,:); % Take the middle row of the unscaled intensity

plot(x,y,'linewidth',2) % Plot the intensity of the middle row of pixels

xlim([2450 2550]) % Set x-axis limit to maximize data readability

xlabel('Length [Pixels]')

ylabel('Intensity [Arb. units]')

title('Horizontal Intensity Cross-Section')

% Plot the vertical cross-section

figure(5)

x=linspace(1,5000,5000);

y=I1(:,2500); % Take the middle column of the unscaled intensity

plot(x,y,'linewidth',2) % Plot the intensity of the middle column of pixels

xlim([2450 2550]) % Set x-axis limit to maximize data readability

xlabel('Width [Pixels]')

ylabel('Intensity [Arb. units]')

title('Vertical Intensity Cross-Section')