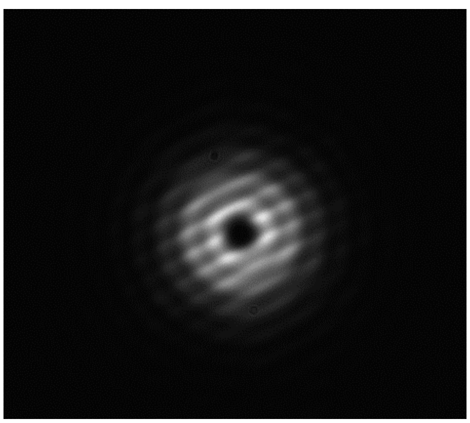
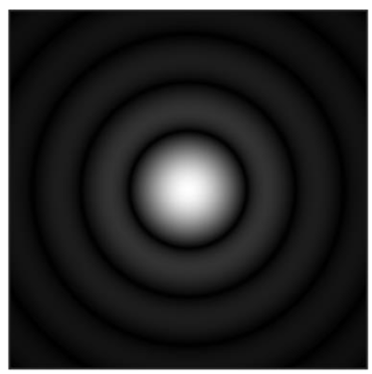
Numerical Simulation of Diffraction Pattern of a Circular Aperture in a Near Field.

I. Introduction

Before spring break we built a beam expander and used it to illuminate different targets to observe diffraction and interference patterns. There was a circular aperture among the targets, and some groups observed a strange diffraction pattern that had a dark spot in the middle instead of the bright spot predicted by Fraunhofer (*far field*) diffraction (the *Airy disk* pattern, Figure 1b). The reason for this is that the Airy disk pattern well-describes the intensity of the diffraction pattern in the far field, but the screen in our experiment for most groups was not far enough away from the aperture.



a)



b)

Figure 1. Diffraction pattern of circular aperture with the dark spot at the center (a), and Airy disk pattern for the circular aperture (b).

The condition of being in the far field for the circular aperture is:

$$d > \frac{(2a)^2}{\lambda} \tag{1}$$

where d is the distance between the aperture and the screen, a is the aperture radius, and λ is the wavelength of the incident light. If you plug values from our lab into (1) you will see that the far field condition is not satisfied in our experiment. Therefore we cannot use the Airy disk function:

$$I(\theta) = \frac{P_a A}{\lambda^2 d^2} \left[\frac{2J_1(ka \sin\theta)}{ka \sin\theta}\right]^2 \tag{2}$$

to describe the intensity I of the diffraction pattern, where P_a is the total power incident on the aperture, A is the aperture area, k is the wave number ($= 2\pi/\lambda$), θ is the angle of observation on the far-away screen, and J_1 is the Bessel function of the first kind of order one.

In this case we should calculate the diffraction pattern intensity numerically, by applying the *Huygens-Fresnel principle*, since we cannot analytically solve for this pattern. This is our goal for this lab: learn about the Huygens-Fresnel principle and use it to numerically simulate the diffraction pattern of a circular aperture in the *near field*.

II. Huygens-Fresnel principle

The aperture in our experiment is illuminated by an expanded laser beam. We can approximate this beam, at least at a small aperture, as a plane uniform wave, with the wave front parallel to the plane of the aperture. According to the Huygens-Fresnel principle we can divide this wave front into small elements dS_a that are themselves light sources (Figure 2) of modified spherical waves that propagate through space according to:

$$A_a(r) = A_0 K(\theta) \frac{e^{ik|r-r_\mathrm{a}|}}{|\boldsymbol{r}-\boldsymbol{r}_\mathrm{a}|} dS_a, \tag{4}$$

where $\boldsymbol{r_a}$ is the vector location of the center of the element dS_a, $\boldsymbol{r}$ is the position vector from the center of this element, and A_0 is the amplitude of the original wave at the center of the element dS_a. (Remember, we consider our wave to be plane and uniform, so all elements have the same wave amplitude A_0). The difference between a pure spherical wave and the modified spherical wave described by equation (4) is the factor $K(\theta)$:

$$K(\theta) = -\frac{i}{2\lambda}(1 + \cos\theta) \tag{5}$$

where θ is the angle between the normal vector of the area element dS_a and the direction of the wave propagation. The wave front of the wave (1) is spherical, the same as for spherical wave, but unlike the spherical wave, the amplitude of the wave (1), depends on the direction of propagation due to the factor $K(\theta)$.

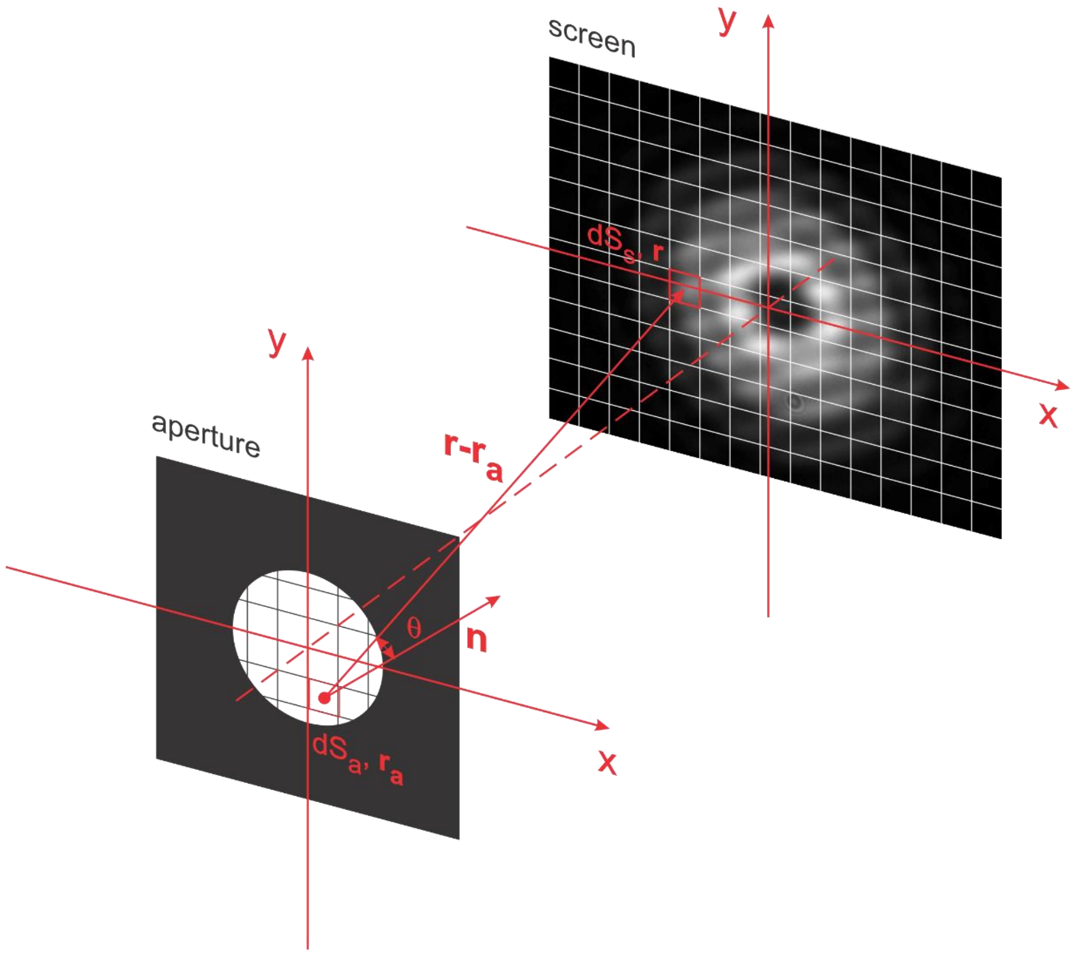


Figure 2. Division of the aperture and the screen into small surface area elements. The wave amplitude at any screen surface element dS_s can be calculated as a sum of the amplitude at dS_s of the waves originated from each of the aperture surface elements dS_a (Equations 6 and 7).

The amplitude $A(r)$ of the resulting wave at any location r is a result of interference among all of the waves produced by all of the aperture elements dS_a and can thus be calculated as an integral over the surface S of the aperture:

$$A(r) = \int_S A_0 K(\theta) \frac{e^{ik|\boldsymbol{r}-\boldsymbol{r}_a|}}{|\boldsymbol{r}-\boldsymbol{r}_a|} dS_a \tag{6}$$

If integral (3) cannot be solved analytically, it can be solved numerically as a sum over all finite elements dS_a:

$$A(r) = \sum_S A_0 K(\theta) \frac{e^{ik|r-r_a|}}{|r-r_a|} dS_a \tag{7}$$

The *intensity* of the total wave at any point is proportional to $|A(r)|^2$:

$$I(r) \propto |A(r)|^2 \tag{8}$$

Intensity is the power ([energy]/[time]) that the wave carries through a unit surface area ([length]^2). The power that wave carries over an entire area $S(r)$ is thus:

$$P_s(r) = I(r)S(r) \propto |A(r)|^2 S(r) \tag{9}$$

Equations (8) and (9) should have some additional constant factors that depend on the nature of the wave, but for all numerical analysis we are not concerned with prefactors, only the dependence of intensity on position on the screen.

III. Simulation

Before applying equation (7) to calculate the diffraction pattern of the circular aperture we have to address a few questions:

- How many elements dS_a will we use to represent the aperture?
- What is the amplitude of the initial wave A_0 at the aperture?
- How many elements dS_s will we use to represent the screen?

Number of aperture elements dS_a

The question of how many elements dS_a to use to represent the aperture is very important. The more elements we use, the better the sum (7) approximates the integral (6). At the same time, the more elements we use, the longer it will take to calculate the sum numerically.

We have to choose an initial number of elements large enough to obtain a reasonable result, but at the same time not very large, so that we will be able to debug the code within a reasonable time frame. After the code is working, we can increase the number of aperture elements until the intensity pattern observed on the screen does not change significantly as element size decreases further; then the element size is small enough to see all details of the pattern given the screen resolution.

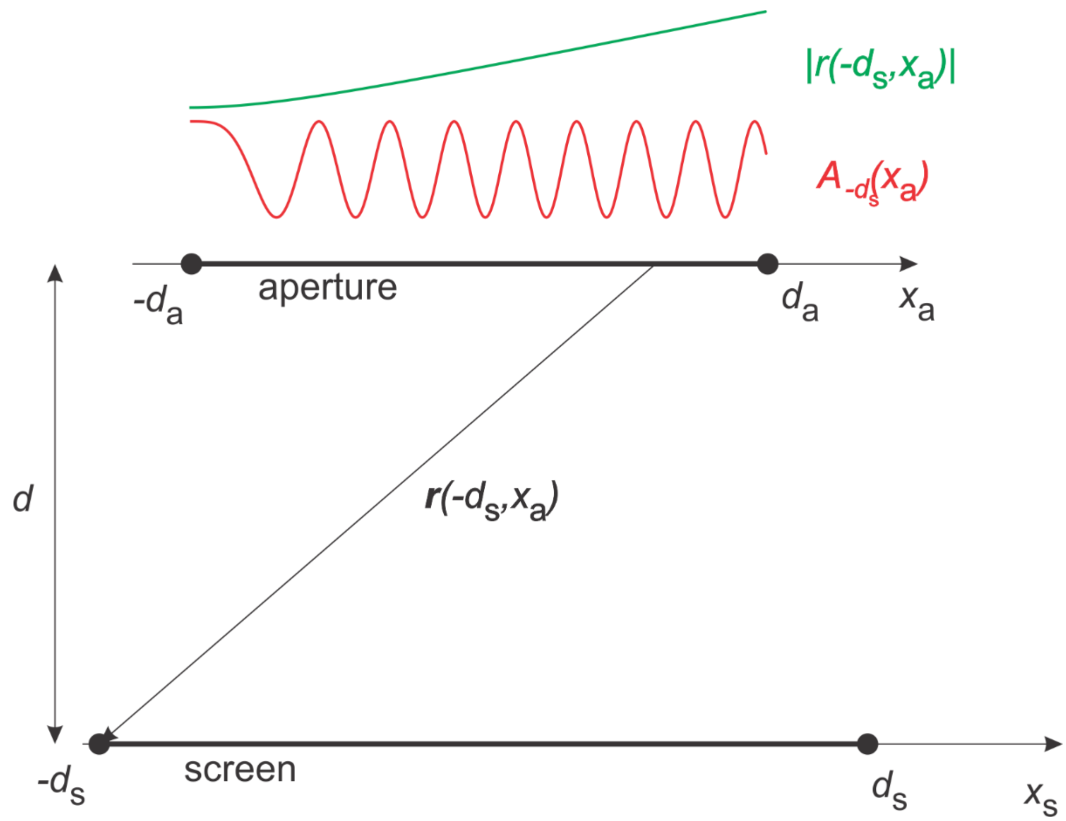


Figure 3. Diagram for finding the initial size of the aperture element dS_a for numerical simulation. See the text below for more details.

To find a reasonable order of magnitude estimate for the initial number of elements dS_a, let's consider a one-dimensional aperture and one-dimensional screen (Figure 3). The aperture is located on the x_a axis, between points $-d_a$ and d_a, and the screen is located on the x_s axis, between points $-d_s$ and d_s. The distance between axes x_a and x_s is equal to d. Wave A propagates from the aperture to the screen.

Let's consider the value of the wave $A_{-d_s}(x_a)$ that originates at point x_a of the aperture and arrives to point $-d_s$ on the screen. The red curve shows the dependence of the amplitude of this wave as a function of wave origin x_a. (We are considering a single instant of time, so temporal oscillations are ignored.) We can see that the larger the distance $r(-d_s, x_a)$ is, shown with the green curve, the faster $A_{-d_s}(x_a)$ oscillates along the axis x_a. To resolve these oscillations when we calculate the sum (7), and therefore resolve interference effects between adjacent screen elements, we need at least two aperture elements per one oscillation of $A_{-d_s}(x_a)$. This means that the difference Δr in $r(-d_s, x_a)$ for these two elements should at least be equal to half of the wavelength of the light. Because $r(-d_s, x_a)$ has a linear dependence on x_a at large x_a, we can well-approximate Δr as:

$$\Delta r \approx \frac{dr(-d_s, x_a)}{dx_a} \Delta x_a \tag{10}$$

where the derivative is evaluated at point $x_a = d_a$. (The two aperture elements we are considering are very close to one another, d_a and $d_a - \Delta x_a$.) **Show that for $\Delta r = \lambda/2$, the distance between two elements should be**:

$$\Delta x_a = \frac{\lambda}{2(d_s + d_a)} \sqrt{(d_s + d_a)^2 + d^2} \tag{11}$$

For a given distance between the elements, the area of each aperture element, going back to our two-dimensional aperture and two-dimensional screen, is therefore:

$$dS_a = (\Delta x_a)^2 \tag{12}$$

The aperture diameter is 0.9 mm.

Amplitude A_0 of the initial wave at aperture

The total power of the beam P_a through the aperture is unknown. We can choose $P_a = 1$, and calculate A_0 using equation (9) since, again, we are concerned only with the functional form of intensity on the screen, not dependence on constants.[1]

Number of screen elements dS_s

There are two ways to estimate the necessary number of screen elements. First, we can use the same approach that we used when we tried to figure out how many elements to use to represent the aperture. In this case, the answer appears to be the same; the initial estimate of distance Δx_s between elements dS_s is

$$\Delta x_s = \frac{\lambda}{2(d_s + d_a)} \sqrt{(d_s + d_a)^2 + d^2} \tag{13}$$

[1] In this way, we specify the "units" of power in the simulation to be equal to 1 unit of power at an aperture element.

Alternatively, we can choose the distance Δx_s to be equal to the distance between pixels of the camera sensor, and keep it constant. The resolution of the camera sensor is 1280×1024 pixels and the pixel size is $4.8 \times 4.8\ \mu m$.

We do *not* have to calculate the amplitude of the total wave $A(r)$ at *all* elements dS_s of the screen. We can use the fact that the diffraction pattern has circular symmetry, and calculate $A(r)$ for the set of elements along the x axis of the screen (Figure 4). The amplitudes for the rest of the elements on the screen could then be found from this data using the distance from the pixels to the center of the screen.

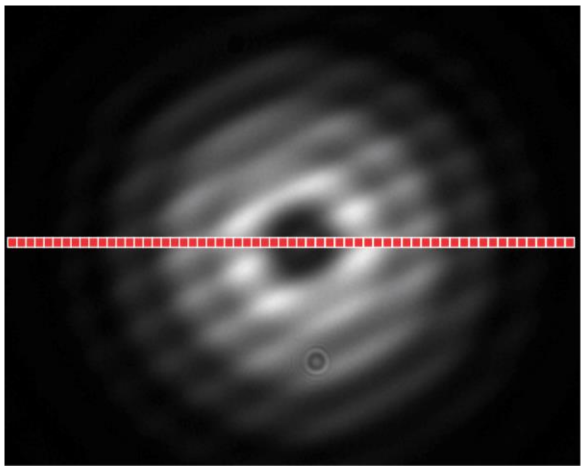


Figure 4. For diffraction pattern with circular symmetry there is no need to run the simulation for all elements on the screen. It is enough to run the simulation for any line of pixels that goes through the center of the pattern, and then this information can used to find the intensity at any other part of the screen by circular symmetry.

IV. Template of the script for numerical simulation of the diffraction pattern

We have created a template of the script that you can use for the simulation. The template consists of five sections and follows section **II** of this manual. It has comments, pieces of code that we think would be difficult for you to write on your own, and parts that you need to fill in. Read the description of the script below before you start to edit it.

Section 1. Variables and parameters

This section is designed to create all variables that are essential for the simulation, such as the laser wavelength, the distance between the aperture and the screen, the aperture radius, and the screen size. It is convenient to have all of them at the same place so that you know where to find them if any change should be made.

You can also use this section to calculate some other variables, such as aperture area.

Section 2. Dividing aperture into area elements

In this section we calculate the initial value *dxa* for the spacing between aperture elements. Later you can adjust *dxa* as needed. This section has a piece of code that was written for you that creates two 2D arrays, *XaM* and *YaM*. Array *XaM* contains x coordinates of all aperture elements, and array *YaM* contains y coordinates of all aperture elements. The structure of *XaM* and *YaM* is shown in Figures 5a and 5b.

XaM

$$\begin{matrix} -3 & -2 & -1 & 0 & 1 & 2 & 3 \\ -3 & -2 & -1 & 0 & 1 & 2 & 3 \\ -3 & -2 & -1 & 0 & 1 & 2 & 3 \\ -3 & -2 & -1 & 0 & 1 & 2 & 3 \\ -3 & -2 & -1 & 0 & 1 & 2 & 3 \\ -3 & -2 & -1 & 0 & 1 & 2 & 3 \\ -3 & -2 & -1 & 0 & 1 & 2 & 3 \end{matrix}$$

a)

YaM

$$\begin{matrix} 3 & 3 & 3 & 3 & 3 & 3 & 3 \\ 2 & 2 & 2 & 2 & 2 & 2 & 2 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -1 & -1 & -1 & -1 & -1 & -1 & -1 \\ -2 & -2 & -2 & -2 & -2 & -2 & -2 \\ -3 & -3 & -3 & -3 & -3 & -3 & -3 \end{matrix}$$

b)

A0M

$$\begin{matrix} 0 & 0 & A_0 & A_0 & A_0 & 0 & 0 \\ 0 & A_0 & A_0 & A_0 & A_0 & A_0 & 0 \\ A_0 & A_0 & A_0 & A_0 & A_0 & A_0 & A_0 \\ A_0 & A_0 & A_0 & A_0 & A_0 & A_0 & A_0 \\ A_0 & A_0 & A_0 & A_0 & A_0 & A_0 & A_0 \\ 0 & A_0 & A_0 & A_0 & A_0 & A_0 & 0 \\ 0 & 0 & A_0 & A_0 & A_0 & 0 & 0 \end{matrix}$$

c)

Figure 5. Structure of (a) *XaM*, (b) *YaM*, and (c) *A0M* arrays. *XaM* contains x coordinates of aperture elements, *YaM* contains y coordinates of aperture elements, and *A0M* contains the wave amplitudes at the aperture elements.

Section 3. Initializing the amplitude of the wave A_0 at the aperture

In this section we create a 2D array *A0M* of wave amplitudes at aperture elements. The structure of the array *A0M* is shown in Figure 5c.

Section 4. Dividing screen into area elements

In this section we create a 1D array *Xs* that contains x coordinates of the screen elements. This array corresponds to the elements that are shown in Figure 4 in red.

Section 5. Constructing diffraction pattern

This section is the main body of the simulation. **In this section you need to**:

1. Calculate the complex amplitude $A(r)$ of the wave and its intensity $I_{wave}(r)$ for every element of the array of screen elements *Xs*;
2. Calculate the Airy disk pattern $I_{Airy}(r)$ for the array *Xs*, and compare it to $I_{wave}(r)$;
3. Create 2D arrays of *x* and *y* coordinates of the screen elements, analogues to *XaM* and *YaM*;
4. Calculate the intensity $I_{wave}(r)$ for all elements on the screen, and plot it as a 2D image (analogue of the picture taken by camera);
5. Calculate the total power of the simulated diffraction pattern; it should be equal to the total power of the wave at the aperture.