Wave Optics based Numerical Simulation of Wave Propagation
EE 528 – Fundamentals of Photonics

Introduction:

Wave Optics is the branch of optics in which the wave nature of light is essential when defining its propagation. In wave optics, light is considered as a wave. This branch of optics deals with the study of various phenomenal behaviors of light like reflection, refraction, interference, diffraction, polarization, etc. When numerical propagation is combined with the concept of the transmittance function for optical elements, it can be used to model many applications. This kind of simulation is known as a 'wave optics simulation'. The wave optics simulation techniques are directly applicable to atmospheric imaging, astronomy, adaptive optics, free-space optical communications, and LADAR (LAser Detection And Ranging).

In this project, we have investigated wave optics-based numerical simulation of wave propagation with an example of propagation after a lens. More specifically, we will observe the focusing of a lens using wave optics simulation as shown in Figure 1.

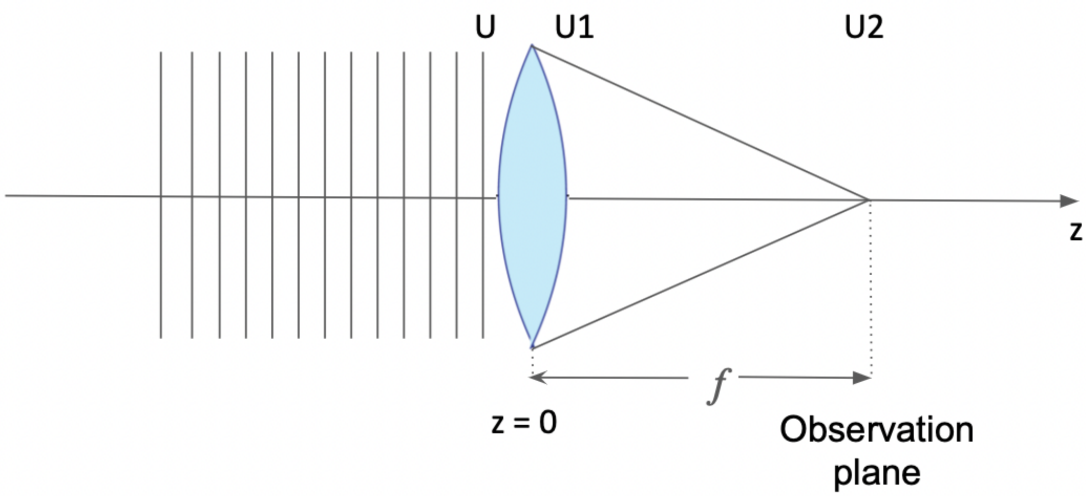

Figure 1: Focusing of a plane wave by a lens

In Figure 1, a plane wave is transmitted through a lens of focal length f. The lens will converge the wave to a point at a distance f from the lens. Hence, we will observe the resulting point image at the observation plane. We consider the lens at z=0 and the field U before the lens can be expressed as $U=Ae^{-jkz}=Ae^{-jk.0}=A$ where A is the amplitude of the plane wave. We can compute the field U1 after the lens using U1=t*U, where t is the transmittance function of the lens. The transmittance function for an ideal, simple lens is given by [1],

$$t_A(x,y) = P(x,y)\exp\left[-j\frac{k}{2f}\left(x^2+y^2\right)\right]$$

where f is the focal length of lens, and P(x,y) is the pupil function.

Pupil function P(x,y) can be expressed as [1]

$$P(x,y) = \text{circ}\left(\frac{\sqrt{x^2+y^2}}{w_L}\right)$$

where circ is the circle function and w_L is the radius of the pupil. Fresnel Transfer Function (TF) Approach is applied to propagate the field *U1* at a distance f from the lens. In Fresnel Transfer Function (TF) Approach, the propagation of light between two planes is regarded as a linear system with impulse response h [2]. Transfer function H is defined as [1],

$$H(f_X, f_Y) = e^{jkz}\exp\left[-j\pi\lambda z\left(f_X^2+f_Y^2\right)\right]$$

where f_X and f_Y are the frequency coordinate, z is the propagation distance between two planes, and λ is the wavelength. At the observation plane, U2 is given by [1],

$$U_2(x,y) = \Im^{-1}\left\{\Im\{U_1(x,y)\}H(f_X,f_Y)\right\}$$

To find the field U2 at the observation plane, we compute the $H(f_x, f_y)$ and multiply it with the Fourier transform of U1. Finally, we did an inverse Fourier transform and find U2. To minimize artifacts in the simulation, we consider critical sampling. The critical sampling expression is defined as [1]

$$\Delta x = \frac{\lambda z}{L}$$

where Δx is sample interval, λ is the wavelength, z is the propagation distance, L is the side length. We can rearrange the critical sampling expression as

$$\boldsymbol{L = \sqrt{M\,\lambda\,z}}$$

where M is the number of samples. Theoretically focusing with lens creates Fraunhofer intensity pattern and the intensity is given by [1],

$$I_2(x,y) = \left(\frac{w^2}{\lambda z}\right)^2 \left[\frac{J_1\left(2\pi \frac{w}{\lambda z}\sqrt{x^2+y^2}\right)}{\frac{w}{\lambda z}\sqrt{x^2+y^2}}\right]^2$$

where J_1 is the Bessel function of the first kind, order 1.

Python code for Simulation:

Focusing with Lens:

We use the number of samples N=500, wavelength λ =0.5 μm, the focal length of the lens, f=1m. The side length is calculated using $L = \sqrt{N\,\lambda\,z}$. The lens is placed at z=0 and the field before the lens U is defined by $U = Ae^{-jkz}$.

```
N=500                           #samples 1D
lambda_= 0.5e-6                 #wavelength
k= (2*pi)/lambda_               #wavenumber
f=1                             #Focal length of lens
L=sqrt(N*lambda_*f)             #spatial grid side length

dx=L/N                          #sample interval
x = np.linspace(-L/2,L/2-dx,N)  #linear coordinates
y = x
X, Y= np.meshgrid(x,x)           #2D cordinates

z=0
A=np.ones(N)
U=A*np.exp(-1j*k*z);             #plane wave field
```

We define pupil function and compute the field U1 after the lens by multiplying U with the transmittance function of the lens, t_lens.

```
w=L/10                    #lens radius
P=circ(np.sqrt(np.square(X)+np.square(Y))/w) #pupil function

t_lens=np.multiply(P,np.exp(-1j*k/(2*f)*np.add(np.square(X),np.square(Y)))) #transmittance function
U1=np.multiply(U,t_lens)       #Field after lens
```

We call the propTF function to propagate the field U1 at a distance f from the lens and compute the intensity of the point image at the observation plane.

```python
U2=propTF(U1,L,lambda_,f)      # Field at a distance f from lens
I_b=np.square(abs(U2))         # Intensity at a distance f from lens
```

We plot the intensity and slice of the intensity of the point image.

```python
fig, axes = plt.subplots(nrows=1, ncols=2,figsize=(22, 6))
img1 = axes[0].imshow(I_b, extent = [min(x),max(x),min(y),max(y)],origin='lower') #display intensity
axes[0].set_xlabel("x (m)",fontsize=16)
axes[0].set_ylabel("y (m)",fontsize=16)

slice_ = int(N/2)
y_ = I_b[slice_,:]
axes[1].plot(x,y_)
axes[1].set_xlabel("x (m)",fontsize=16)
axes[1].set_ylabel("y (m)",fontsize=16)

plt.show()
```

Fresnel Transfer Function (TF) Approach:

The propTF function takes the source field u1 as input and provides an observation field u2 at a distance of z. At first, transfer function H is computed. Then fast fourier transformation (fft) of the source field is computed and multiply it with H. By taking the inverse fft of the multiplication product, observation field u2 is computed.

```python
def propTF(u1,L,lambda_,z):
    # propagation - transfer function approach
    # assumes same x and y side lengths and
    # uniform sampling
    # u1 - source plane field
    # L - source and observation plane side length
    # lambda - wavelength
    # z - propagation distance
    # u2 - observation plane field
    M,N =u1.shape                               #get input field array size
    dx=L/M                                      #sample interval
    k=(2*pi)/lambda_                            #wavenumber
    fx=np.arange(-1/(2*dx),-1/(2*dx)+(M)*1/L,1/L) #freq coords
    FX,FY=np.meshgrid(fx,fx)
    H=np.exp(-1j*pi*lambda_*z*np.add(np.square(FX),np.square(FY)))  #trans func
    H=np.fft.fftshift(H)                        #shift trans func
    U1=np.fft.fft2(np.fft.fftshift(u1))         #shift, fft src field
    U2=np.multiply(H,U1)                        #multiply
    u2=np.fft.ifftshift(np.fft.ifft2(U2))       #inv fft, center obs field
    return u2
```

pupil & jinc function:

circ function is used to define the pupil of the lens.

```
def circ(r):
    out=(np.where(abs(r)<=1, 1, 0))
    return out
```

jinc function computes J1(2pi*x)/x, where J1 is the Bessel function of the first kind, order 1. A masking approach is used to avoid the divide-by-zero condition when x = 0.

```
def jinc(x):
    #jinc function
    #evaluates J1(2*pi*x)/x with divide by zero fix
    #locate non-zero elements of x
    index = np.nonzero(x == 0)
    x1=(np.where(x==0, -1, x))
    #compute output values for all other x
    out = np.divide(special.jv(1, 2*pi*x1),x1)
    out[index] = pi
    return out
```

Theoretical Result:

We also compute the theoretical intensity of the point image from the Fraunhofer Intensity Pattern. We plot the slice of the intensity of the point image for both simulation and theoretical result at a distance f from the lens.

```
a1 = np.square(w)/(lambda_*f)
a2_ = (w/(lambda_*f))*np.sqrt(np.add(np.square(X),np.square(Y)))
a2 = jinc(a2_)
I2_th = np.square(a1)*np.square(a2)
```

```
fig, axes = plt.subplots(nrows=1, ncols=2,figsize=(22, 6))

img1 = axes[0].imshow(I2_th, extent = [min(x),max(x),min(y),max(y)],origin='lower') #display intensity
axes[0].set_xlabel("x (m)",fontsize=16)
axes[0].set_ylabel("y (m)",fontsize=16)

slice_ = int(N/2)
y1_ = I_b[slice_,:]
y_ = I2_th[slice_,:]
axes[1].plot(x,y1_)
axes[1].plot(x,y_,'.')
axes[1].set_xlabel("x (m)",fontsize=16)
axes[1].set_ylabel("y (m)",fontsize=16)
axes[1].legend(['Simulation', 'Theoretical'],fontsize = 12)

plt.show()
```

Result Analysis:

We observe the intensity of the point image at a distance f from the lens (observation plane). Figure 2, 3, and 4 show the intensity and slice of the intensity of the point image at a distance f from the lens for w=L/40, w=L/20, and w=L/10 respectively, where w is the radius of the pupil and L is the side length. We have found that with the increment of w, the focus point of the intensity pattern gets smaller, and the slice of the intensity is shrinking as shown in Figure 2, 3, and 4. To get the perfect focus point, we need to change the pupil radius gradually as the diffraction of light is considered in wave optics.

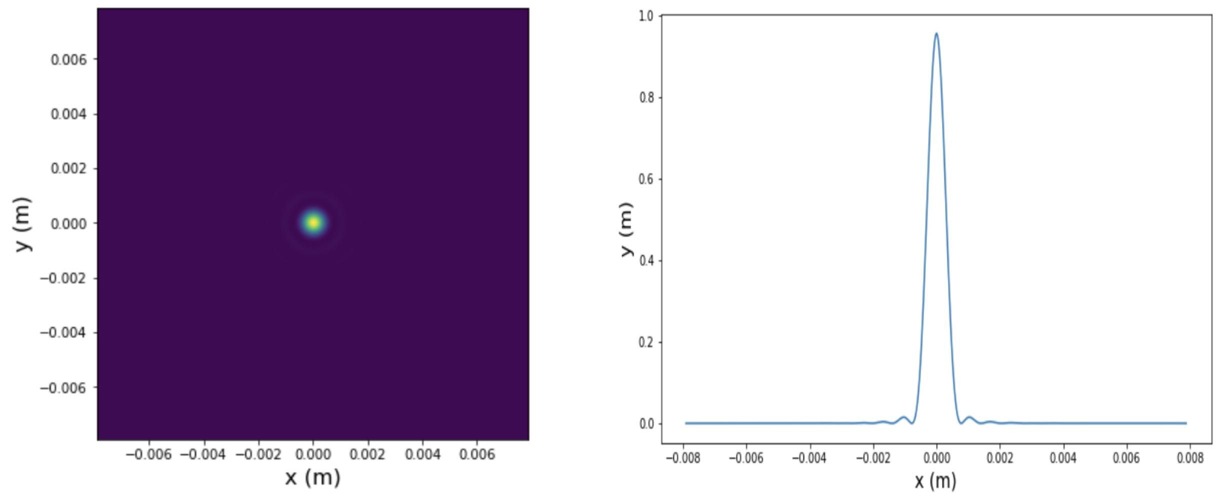

Fig-2: (a) Intensity (b) Slice of Intensity at a distance f from lens for w = L/40

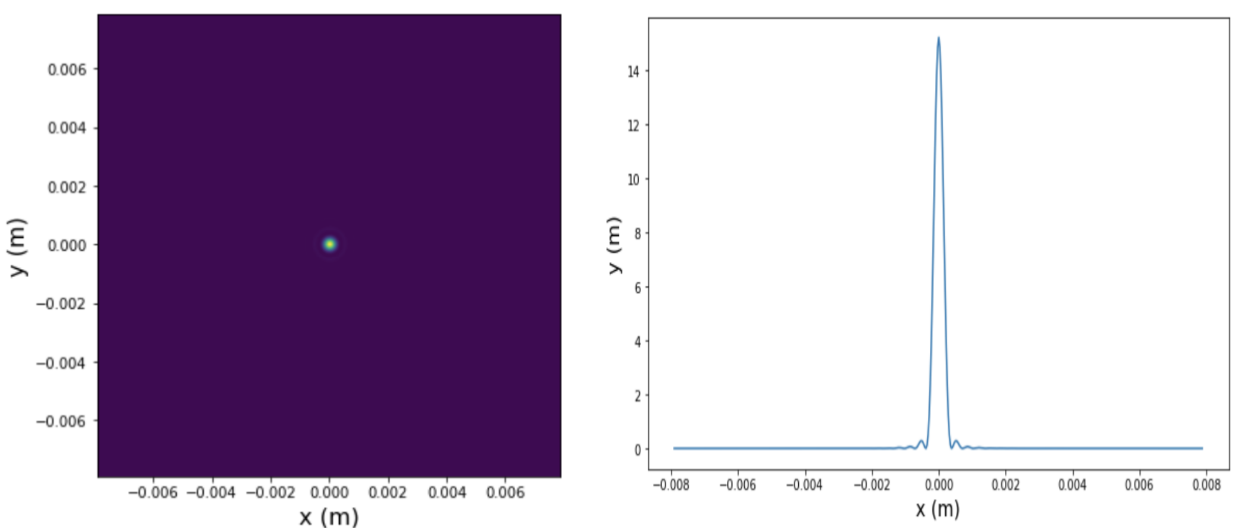

Fig-3: (a) Intensity (b) Slice of Intensity at a distance f from lens for w = L/20

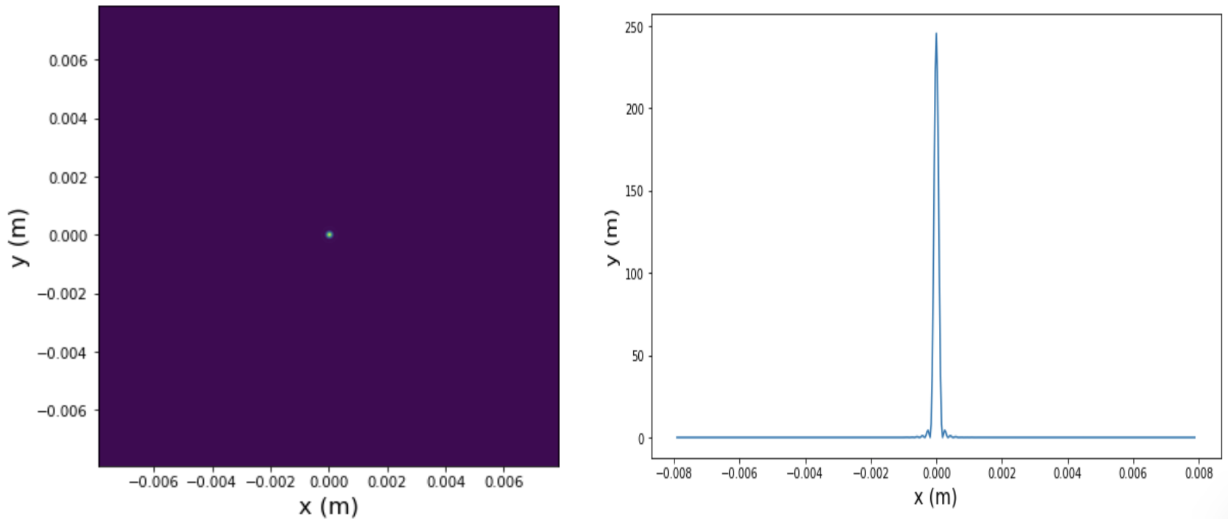

Fig-4: (a) Intensity (b) Slice of Intensity at a distance f from lens for w = L/10

We compare our simulation result with the theoretical result. We have found that the theoretical result matches the simulation result as shown in Figure 5.

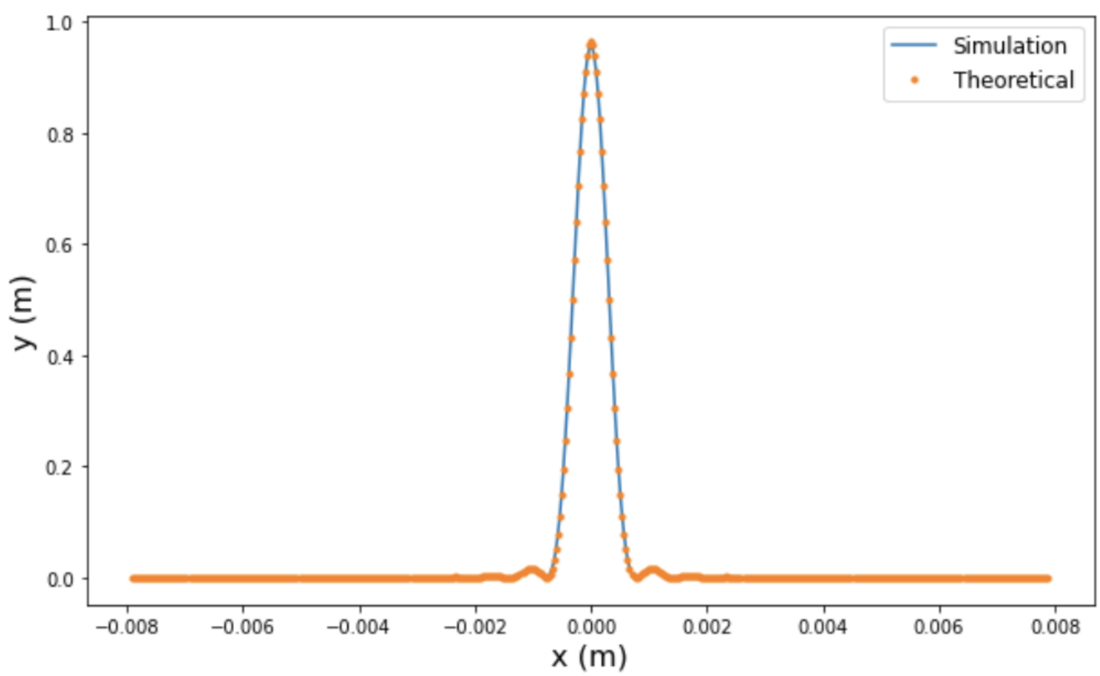

Fig 5: Comparison of intensity slices of Theoretical & Simulation Result

Conclusion:

We study wave optics simulation and apply it for focusing with a lens. We investigate Fresnel Transfer Function (TF) Approach for propagating a field. From critical sampling expression, we find that side length $L = \sqrt{N \lambda z}$, where N is the number of samples, λ is the wavelength, z is the propagation distance. We observe that with the increment of pupil radius, the focus point on the observation plane gets smaller and diffraction is considered in wave optics. Comparison between theoretical and simulation results shows a similar intensity pattern.

References:

1. David Voelz. Computational Fourier Optics: A MATLAB Tutorial 2011
2. Saleh, B. E. A, and M. C. Teich. Fundamentals of Photonics (2nd ed.). New York, Wiley, 2007.