DOING PHYSICS WITH PYTHON

COMPUTATIONAL OPTICS

RAYLEIGH-SOMMERFELD 1

DIFFRACTION INTEGRAL

BEAM PROPAGATION

UNIFORMLY ILLUMINATED CIRCULAR APERTURE

Ian Cooper

Please email me any corrections, comments, suggestions or additions: **matlabvisualphysics@gmail.com**

DOWNLOAD DIRECTORIES FOR PYTHON CODE

Google drive

GitHub

emRS02.py Irradiance in XY planes

emRS02Z.py Irradiance along the +Z axis

INTRODUCTION

The **Rayleigh-Sommerfeld diffraction integral of the first kind** is used to calculate the intensity from a circular aperture that is uniformly illuminated by monochromatic light of wavelength λ.

The geometry of the aperture and observation spaces is shown in figure 1 and figure 2 shows an outline of how to the RS1 diffraction integral is computed in Python. Figure 3 shows a [2D] and [3D] view of the uniform aperture intensity.

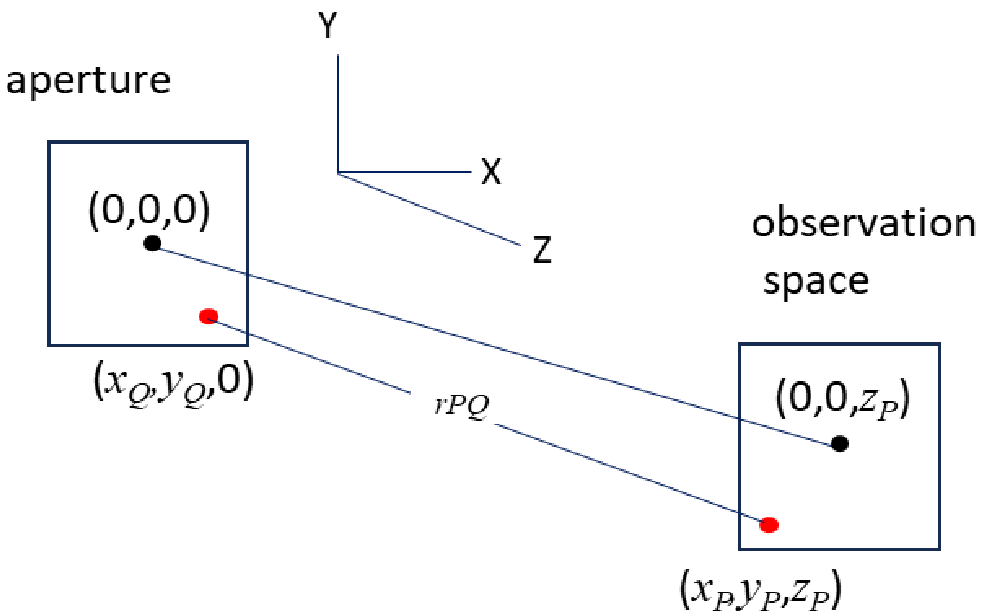


Fig. 1. Geometry of the aperture and observation spaces.

Rayleigh-Sommerfeld diffraction integral

$$E_P = \frac{1}{2\pi}\iint_{S_A} E_Q \frac{e^{jk\,r_{PQ}}}{r_{PQ}^{\;3}} z_p (j\,k\,r_{PQ} - 1)\;\mathrm{d}S$$

↓ numerical integration: [2D] Simpson's 1/3 rule

$$E_P(x_P,y_P,z_P) = z_P \sum_{m=1}^{n_Q}\sum_{n=1}^{n_Q}\left(\left(\frac{e^{jk\,r_{PQmn}}}{r_{PQmn}^{\;3}}\right)\left(jkr_{PQmn}-1\right)\left(E_{Qmn}S_{mn}\right)\right)$$

nPxnP matrix **EP**

nQxnQ matrix **MP**

nQxnQ matrix **EQ**

nQxnQ matrix **S**

Fig. 2. Matrices used in computed the diffraction integral.

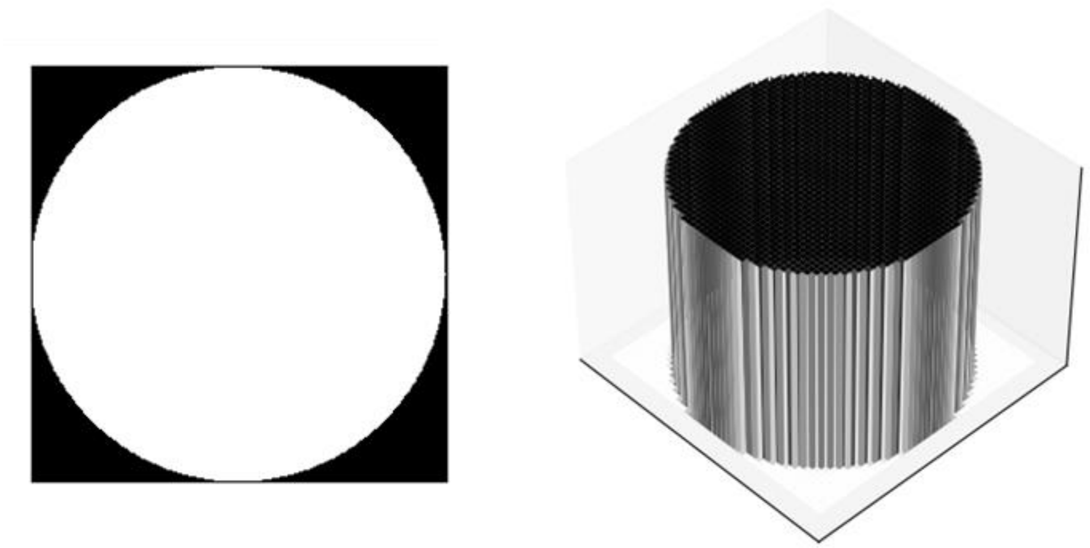


Fig. 3. [2D] and [3D] views of the uniform aperture intensity.

emRS102.py

Both the Fraunhoffer and Fresnel diffraction patterns in the observation space can be computed easily by evaluating the RS1 diffraction integral.

The **Fraunhofer diffraction** pattern for the circular aperture is circularly symmetric and consists of a bright central circle surrounded by series of bright rings of rapidly decreasing strength between a series of dark rings (figure 4). The bright and dark rings are not evenly spaced. The bright central region is known as the **Airy disk**.

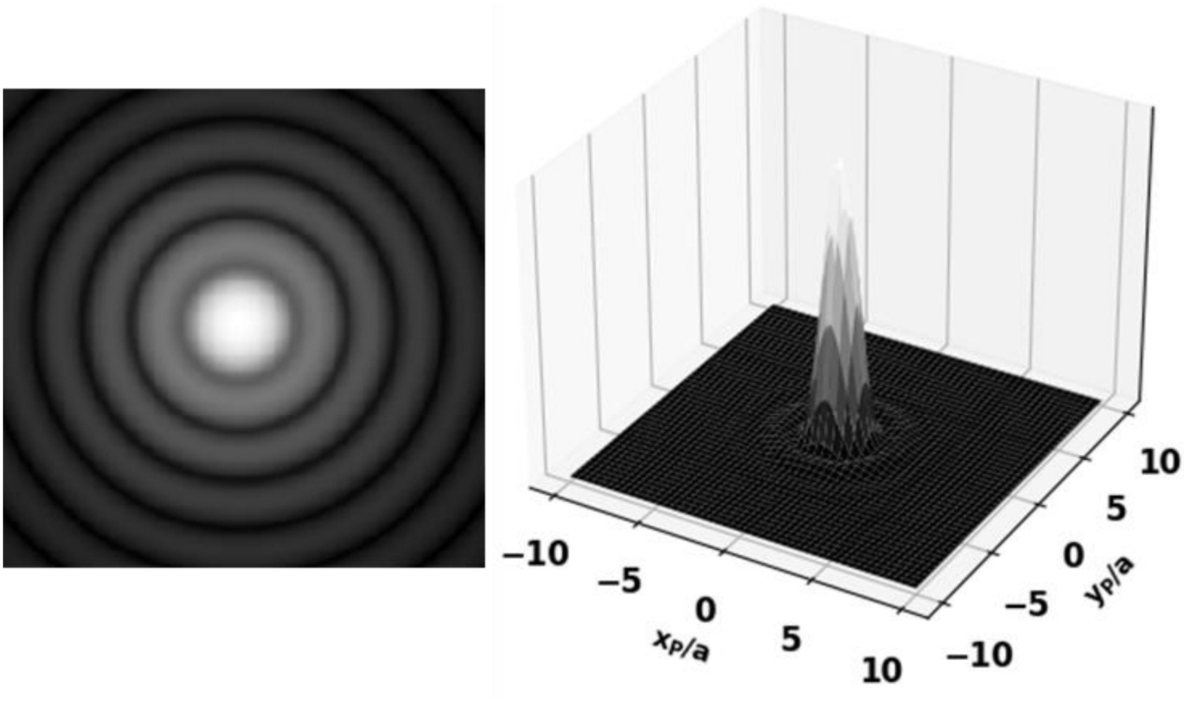


Fig. 4. Fraunhoffer diffraction pattern of a circular aperture of radius a. **emRS102.py**

The image shown in figure 4 is like a black and white time exposure photograph of the diffraction pattern that would be observed on a screen for a uniformly illuminated circular aperture. The bright centre spot corresponds to the zeroth order of diffraction and is known as the Airy disk and It extends to the first dark ring.

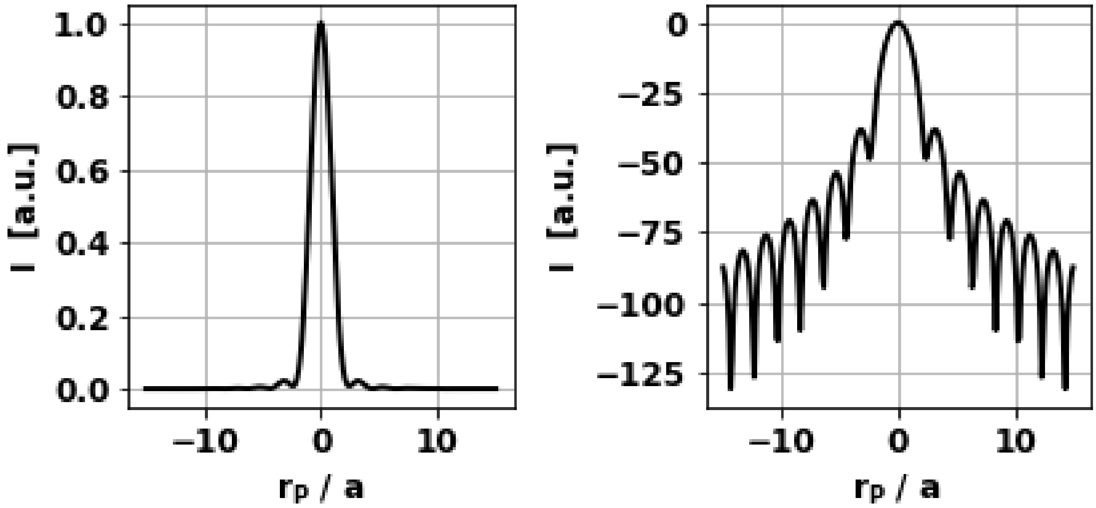


Fig. 4. The irradiance patterns in a radial direction (linear scale and decibel scale for the irradiance). **emRS102.py**

The input parameters for figure 4 are

```
#%% INPUT PARAMETERS
# Grid points: Q aperture space    nQ format odd number
#            P observation space nP format integer * 4 + 1
nQ = 199; num = 59;   nP = num*4+1
# Wavelength [m]
wL = 632.8e-9
# Aperautre space: radius a / XY dimensions of aperture [m]
a = 4e-4; aQx = 2*a; aQy = 2*a
# Observation spacehalf: half-width % zP   [m]
xPmax = 10*a; yPmax = 10*a; zP = 1
```

In the far-field or Fraunhofer region, the irradiance is given by equation 1

(1) $$I = I_o \left(\frac{J_1(v_P)}{v_P} \right)^2$$ **Fraunhofer diffraction only**

where J_1 is the Bessel function of the first kind and v_p is the radial optical coordinate and is a scaled perpendicular distance from the optical axis (equation 2).

(2) $$v_P = \frac{2\pi}{\lambda} a \sin\theta \qquad \sin\theta = \frac{\sqrt{x_P{}^2 + y_P{}^2}}{\sqrt{x_P{}^2 + y_P{}^2 + z_P{}^2}}$$

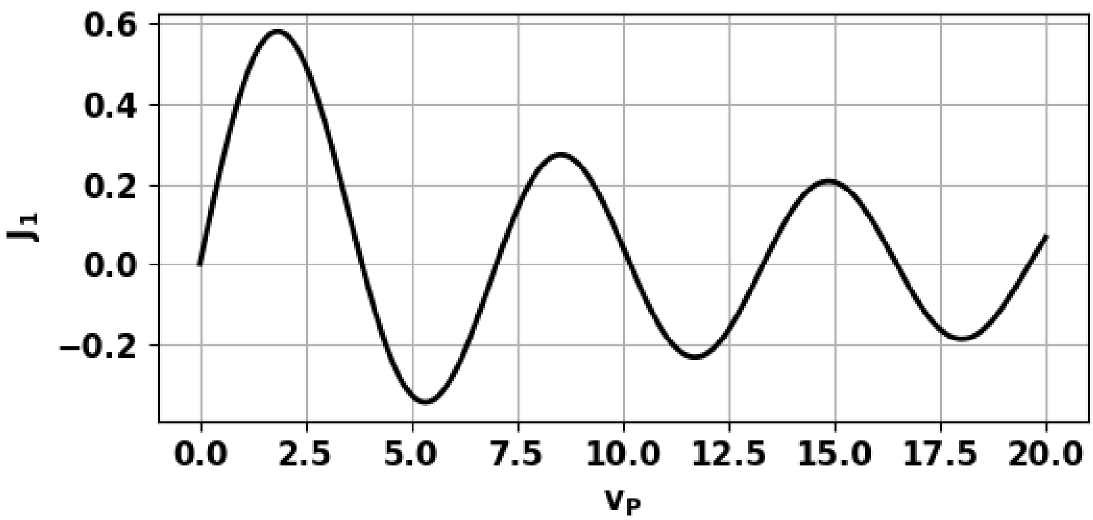


Fig. 5. Bessel function of the first kind using a Python function.

emRS102.py

```
from scipy.special import
j1   num = 9999
v = linspace(0,20,num)
J1 = j1(v).
```

The zeros of the Bessel function are given in Table 1. The row A are values from the internet and row P are calculated in Python using a zero crossing routine.

Table 1. Zeros of the Bessel function J1

A	3.832	7.016	10.174	13.324	16.471	19.616
P	3.831	7.015	10.172	13.323	16.469	19.614

```
# Radial optical coordinates for zeros in Bessel function
J1index = zeros(10); p = 0
for c in range(num-2):
   q = J1[c]*J1[c+1]
   if q <= 0:
     J1index[p] = c
     p = int(p+1)
J1index = J1index.astype(int)
vZeros = v[J1index]
```

In figure 4, the radial coordinate is r_P / a. The zeros in the irradiance are found with the Python function **find_peaks** which is used to find the minimum values in the irradiance function expressed in decibels.

```
# RS1 predictions for location radial positions for zero intensity
#   Radial intenisty array  IX and IdB
IdB = 10*np.log10(Ix)
q = find_peaks(-IdB)
xZ_RS1 = xP[q[0]]/a
```

The zeros in the radial coordinate r_P / a are also found from the Table 1, row A (Bessel function zeros).

```
# Observation space: radial positions for zero intensity xZ
#   Zeros for Bessel function of first kind  J1(rho = 0)
```

```
#   Angles (theta) for zeros in diffraction pattern
rho =
np.array([3.8317,7.0156,10.1735,13.3237,16.4706,19.6159])
theta = np.arcsin(rho/(a*k))
xZ = zP*np.tan(theta)/a
```

The zeros in the irradiance are given in Table 2. The row A are values from the internet Bessel function zeros and row P are calculated in Python.

Table 2. Zeros of the irradiance: radial coordinate r_P / a.

A	2.41	8.39	10.37	12.35		
P	2.42	8.39	10.30	12.33	14.36	19.614

We can estimate the strengths of the irradiance peaks again using the find_peaks function

```
# Relative intensities of maxima
q = find_peaks(Ix)
peaks = Ix[q[0]]

→  1.0000   0.0220   0.0046   0.0017   0.0008   0.0005   0.0003
```

Energy enclosed within the dark rings of the diffraction pattern

It is possible to calculate the energy enclosed within a ring of a specified radius on the observation screen by numerically integrating the irradiance with ever increasing radius.

```
# Power enclosed with a circle [a.u.]
r  = xP[indexXY:nP]
Ir  = Ix[indexXY:nP]
Pr = zeros(len(r))
for c in range(len(r)):
   if c > 1:
      Pr[c] = simps(r[0:c]*Ir[0:c],r[0:c])
Pr = 100*Pr/max(Pr)
```

Figure 6 shows the power as a total of maximum power enclosed within circles of increasing radius. About 84% of the energy from the aperture to the observation screen is enclosed within the Airy disk as indicated in figure 6.

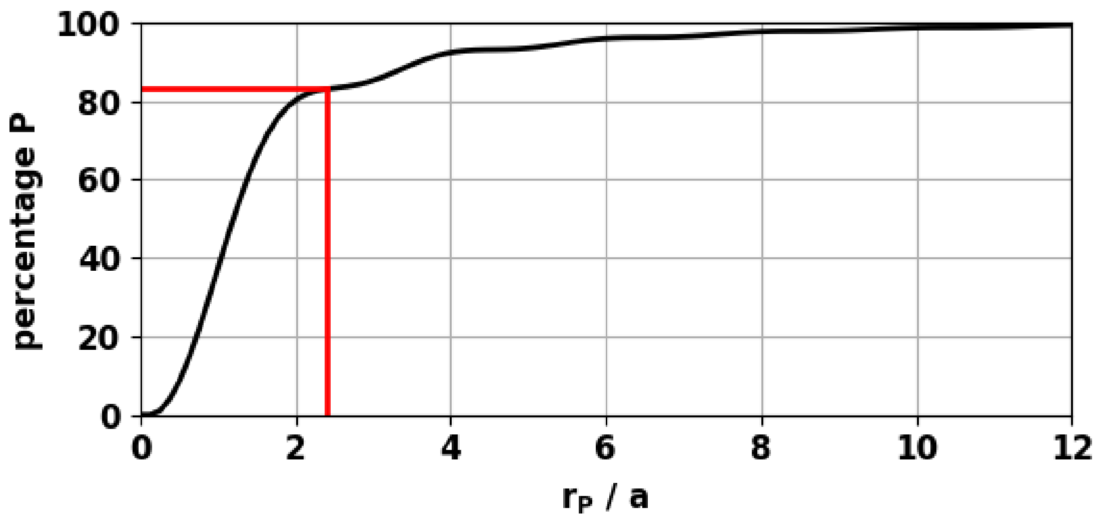


Fig. 6. Percentage power enclosed within rings of increasing radius on the observation screen in the far field for a uniformly illuminated circular aperture. About 84% of the total power is enclosed within the first dark ring. **emRS102.py**

Divergence of the beam and wavelength

The wavelength can be changed in the Python Code **emRS102.py** to investigate the dependence of the irradiance pattern in an XY plane with wavelength (figure 7).

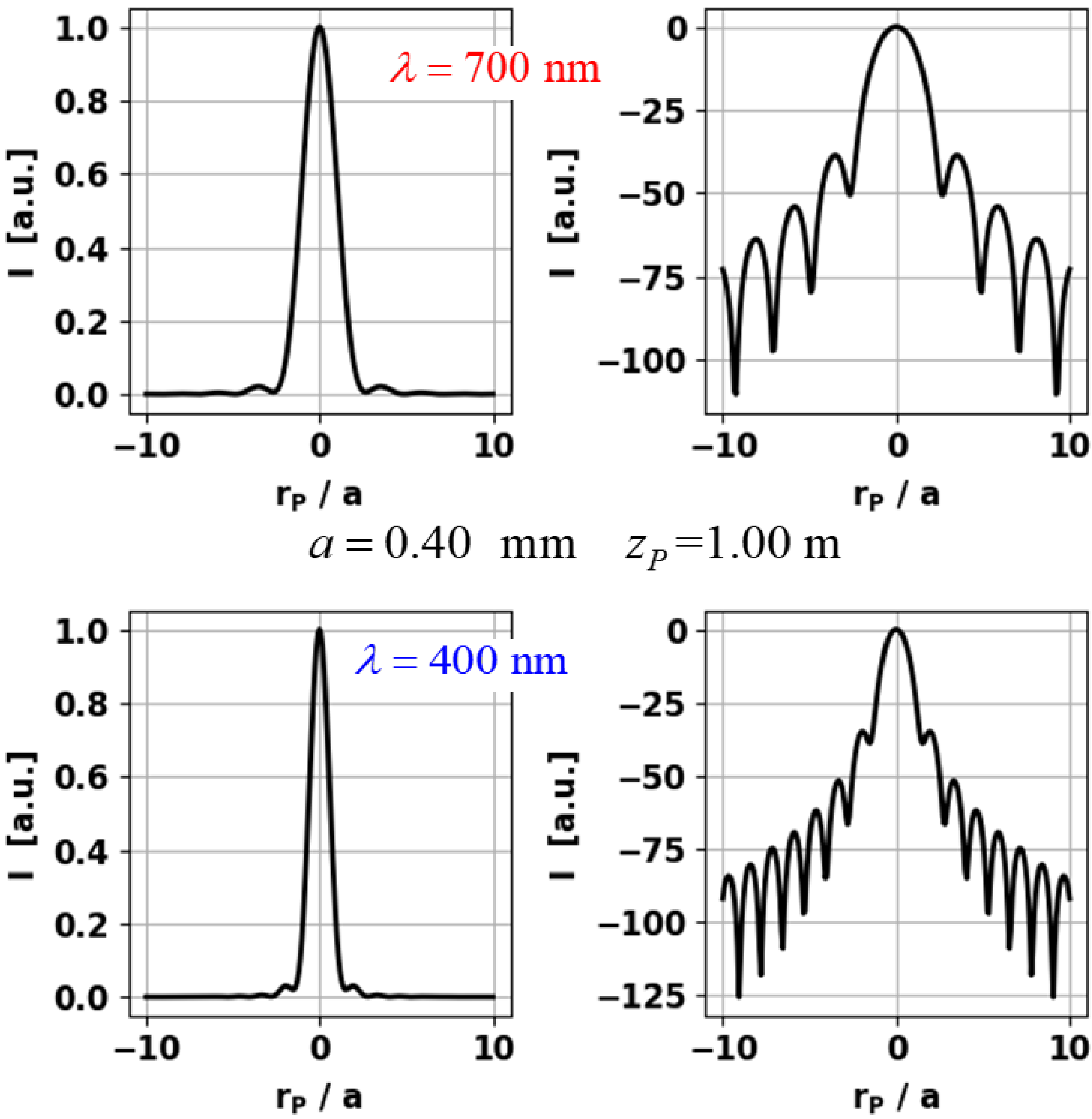
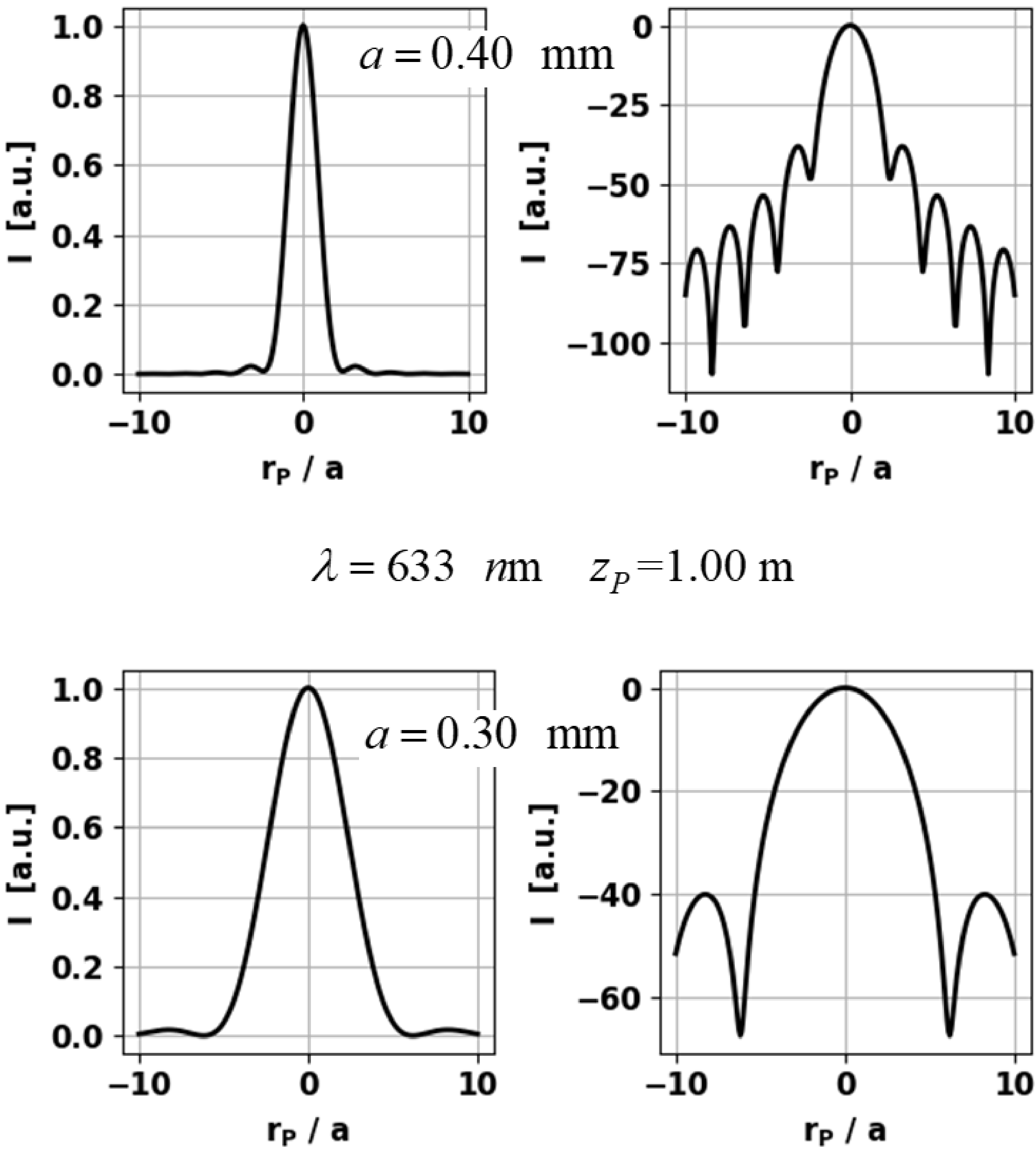


Fig. 7. The smaller the wavelength of the incident radiation, then the narrow the beam. For the blue light, the dark rings are much closer together than for the red light. **emRS102.py**

Divergence of the beam and aperture radius

The radius of the aperture can be changed in the Python Code **emRS102.py** to investigate the dependence of the irradiance pattern in an XY plane with aperture size (figure 8).



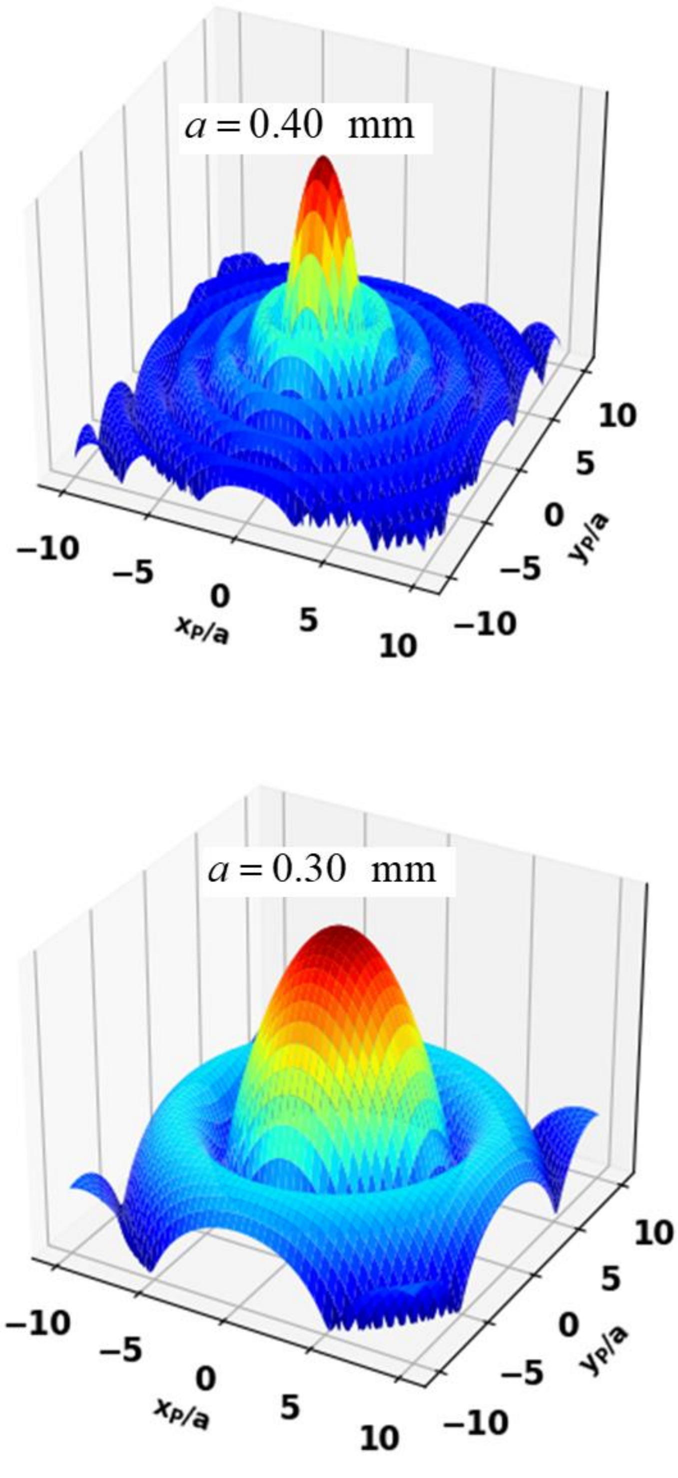


Fig. 8. The smaller the aperture radius, then the wider the beam. For the larger radius aperture, the dark rings are much further apart.

emRS102.py

Fraunhoffer to Fresnel diffraction

The Rayleigh-Sommerfeld diffraction integral of the first kind (figure 2) is valid right up to the aperture for the calculation of the electric field at an observation point P. The transition from Fraunhofer diffraction to Fresnel diffraction can be expressed in terms of the Rayleigh distance. The **Rayleigh distance** in optics is the axial distance from a radiating aperture to a point an observation point P at which the path difference between the axial ray and an edge ray is λ / 4. A good approximation of the Rayleigh distance d_{RL} is

$$d_{RL} = \frac{4a^2}{\lambda}$$

where a is the radius of the aperture. The Rayleigh distance is approximately the value of z_P where the first minimum in the irradiance is greater than zero (figure 9).

$z_P < d_{RL}$ Fresnel diffraction

$z_P > d_{RL}$ Fraunhofer diffraction.

For figures 9 and 10

$$a = 0.40 \text{ mm} \quad \lambda = 633 \text{ nm} \ \rightarrow \ d_{RL} = 1.01 \text{ m}$$

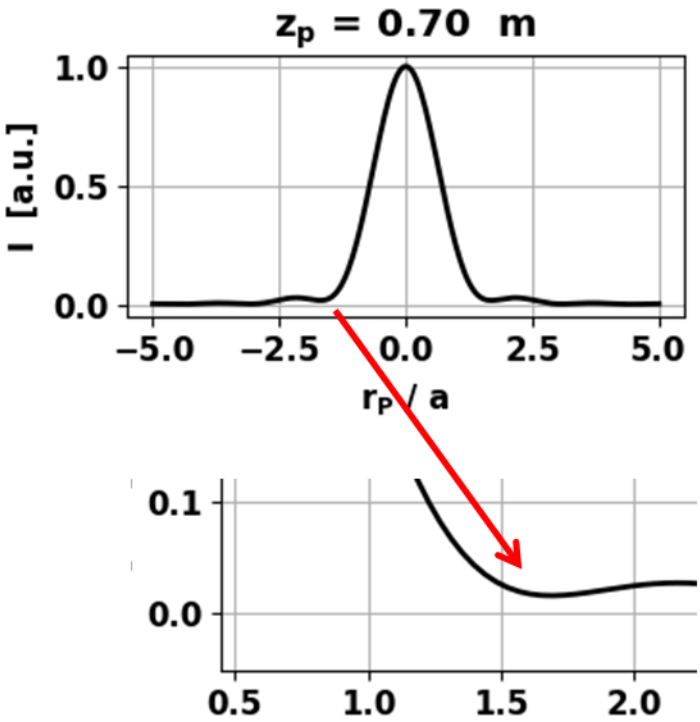


Fig. 9. The first minimum is greater than zero $\left(z_P < d_{RL} = 1.01 \text{ m}\right)$.

emRS102.py

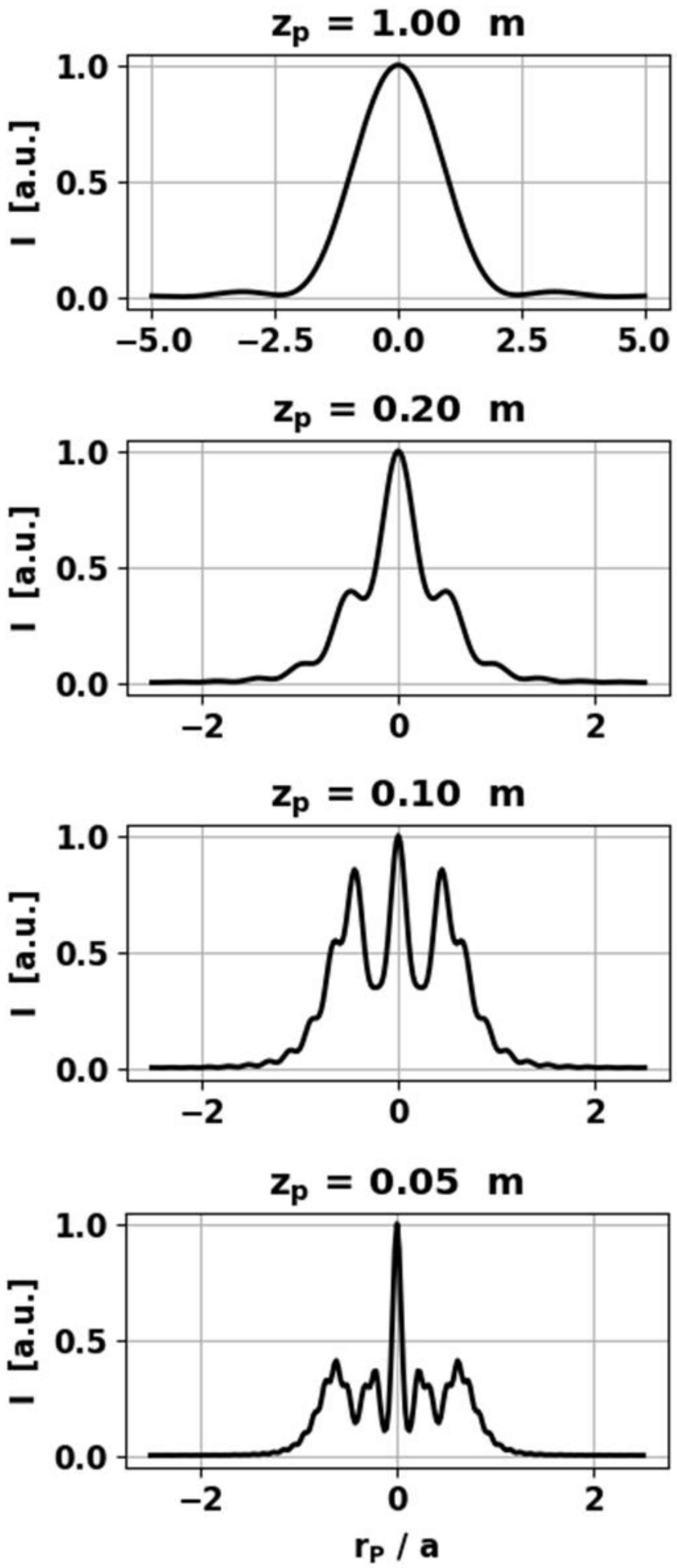


Fig. 10. Transition from Fraunhofer diffraction to Fresnel diffraction.
emRS102.py

Irradiance I_z along the optical axis (+Z axis)

The Python Code **emRS1Z.py** is used to calculate the irradiance as a function of the distance from the centre of the aperture along the +Z axis. THE Rayleigh-Sommerfeld diffraction integral can calculate electric field up to the aperture. Sample computations are shown in figure 11 for Fraunhoffer diffraction) $\left(z_P > d_{RL} = 1.01 \text{ m}\right)$ and figure 12 for Fresnel diffraction $\left(z_P < d_{RL} = 1.01 \text{ m}\right)$.

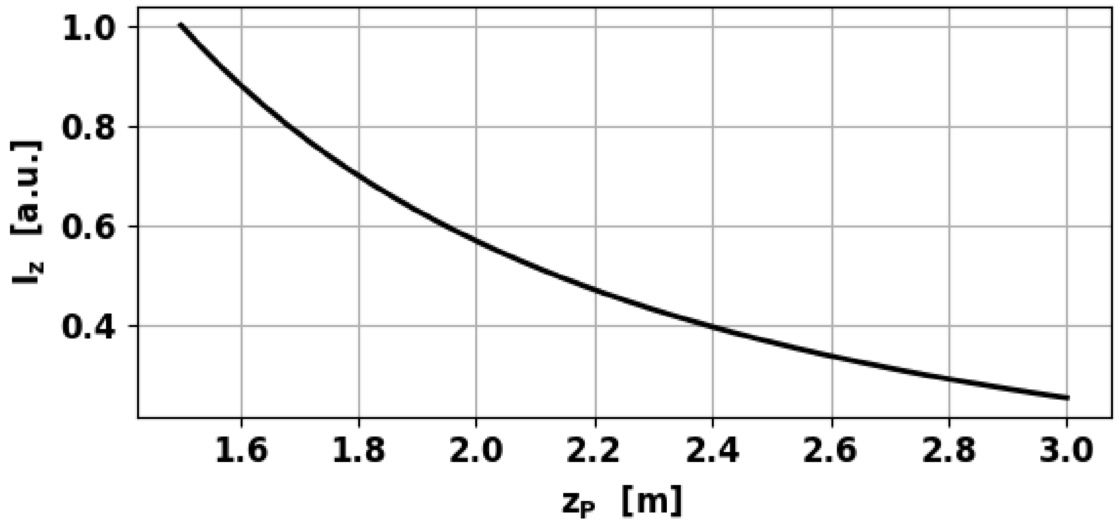


Fig. 11. Fraunhoffer region. The irradiance falls off according to the inverse square law as the light from the aperture can be approximated as a point source. **emRS102Z.py**

nQ = 99 nP = 637
wavelength wL = 633 nm
aperture radius a = 0.400 mm
z1 = 1.50 m z2 = 3.00 m
Iz(z1) = 1.000 Iz(z2) = 0.254 Iz(z2)/Iz(z1) = 0.254

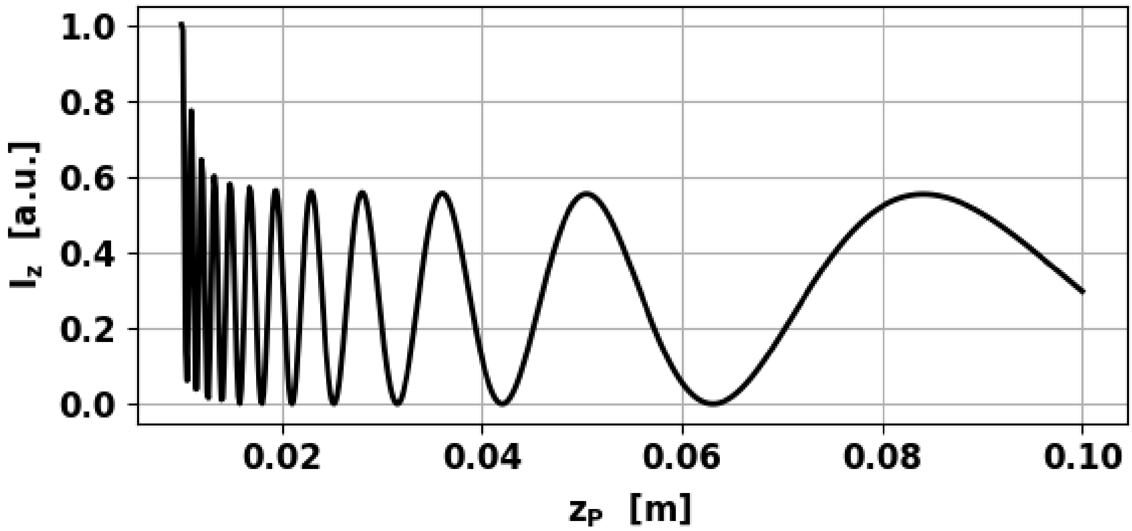


Fig. 12. Fresnel region. The irradiance oscillates near the region of the aperture as the distance z_P increases due to the constructive and destruction effects of the light coming from different regions of the aperture. **emRS102.py**

nQ = 99 nP = 637

wavelength wL = 633 nm

aperture radius a = 0.400 mm

z1 = 0.01 m z2 = 0.10

Iz(z1) = 1.000 Iz(z2) = 0.297 Iz(z2)/Iz(z1) = 0.297