

Computational Optical Imaging

Lecture 1b

Free space propagation

Computational Optical Imaging

- A. Graded weekly exercise sessions (60% of the grade)
- B. Final (40% of the grade)
- C. The graded will be based on the top 10 out 13 graded exercises
- D. No weekly homework
- E. 4 hours per week, 13 weeks
 - 1. First hour (Friday 10:15 to 11:00): Lecture introducing the topic
 - 2. Second hour (Friday 11:15 to 12:00) : Lecture and code on topic
 - 3. Third hour (Friday 12:15 to 13:00): In class exercise on topic
 - 4. Fourth hour (Wednesday 10:15 to 11:00): Graded exercise on topic

Course website

All information, lecture notes, presentations, homework, and announcement for the course will be available on:

<http://moodle.epfl.ch>

course ID: MICRO-421

password for resources: [imaging](#)

Topics

Optical wave propagation

Free space propagation

Angular spectrum and Beam propagation method

Thin transparencies-Lenses and gratings

Imaging

Holography

Multi-layer networks

U-Net

Microscopy

DNN - Unet superresolution

DNN - Unet digital staining

DNN - Phase from Intensity

Endoscopy

DNN for focusing and imaging through MMFs

Matrix method

Scattering media

Phase conjugation

Ptychography

Optical diffraction tomography

Inverse scattering-MaxwellNet

Plan for today

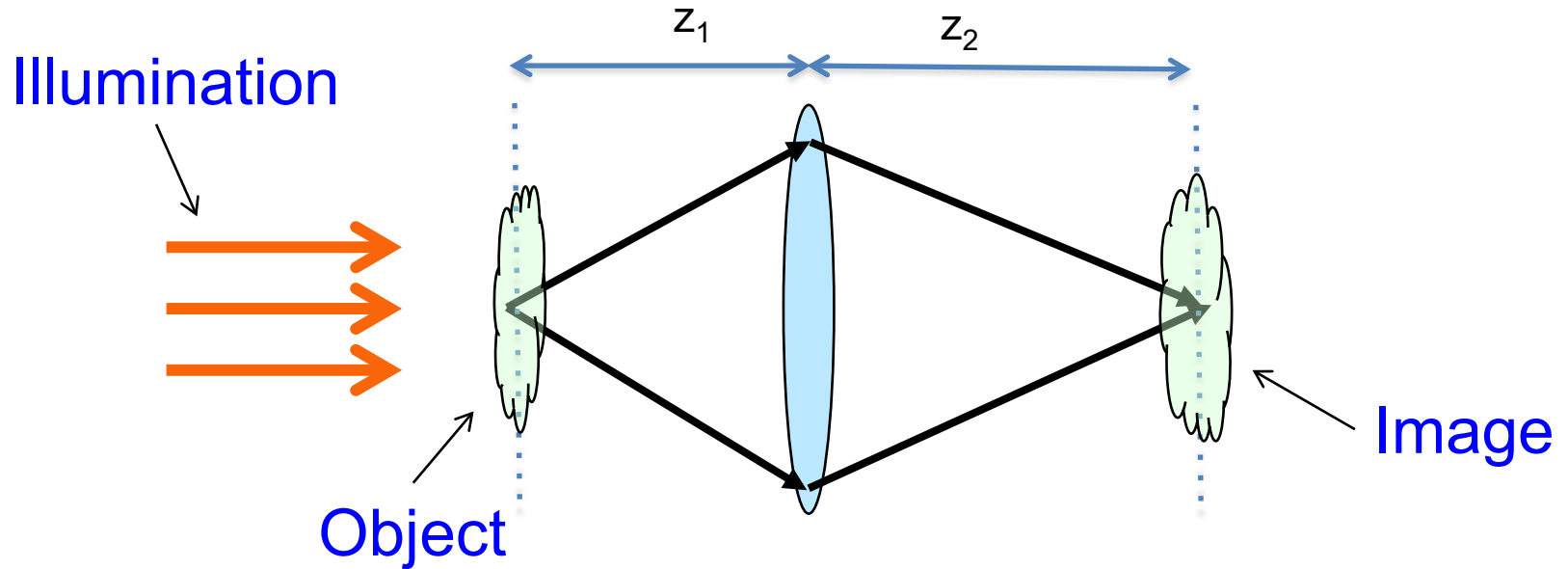
1. Plane waves and the angular spectrum
2. 2D Fourier transforms
3. Examples of free space propagation
4. Code for angular spectrum

Just after sunset





Imaging System



$$\frac{1}{F} = \frac{1}{z_1} + \frac{1}{z_2} \quad M = -\frac{z_2}{z_1}$$

Propagation in Free Space



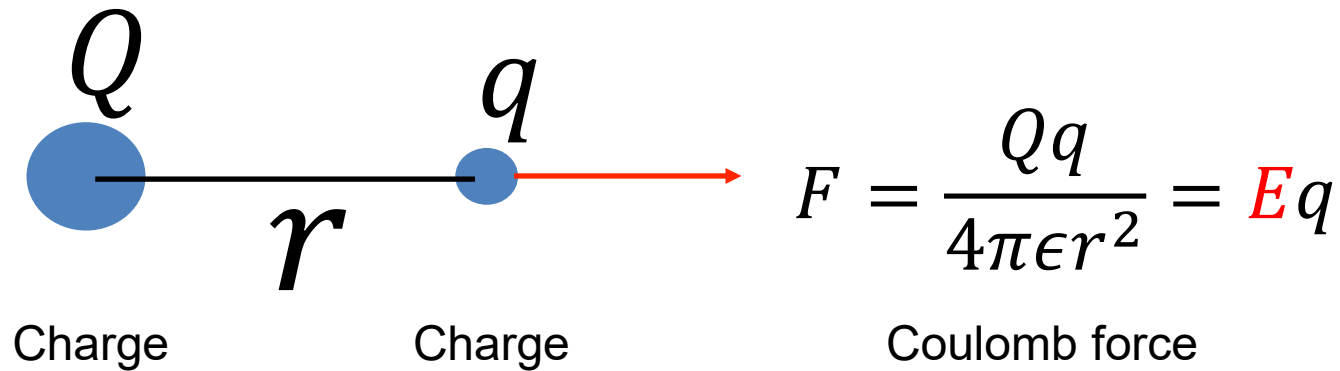
$$\phi(x,y,z) e^{j\omega t} e^{-jkz}$$

Where

$$A(x, y, z) = |E(x, y, z)| e^{j\phi(x,y,z)} e^{+jkz}$$

$$\begin{aligned} k &= \frac{2\pi}{\lambda} \\ \omega &= 2\pi\nu \\ c &= \lambda\nu \end{aligned}$$

Electric Field (E in Volts/m)



Charge Q Charge q

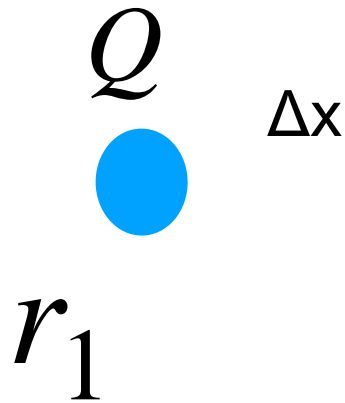
r

$F = \frac{Qq}{4\pi\epsilon r^2} = E q$

Coulomb force

F and E are vectors

Charges moving at Constant Velocity



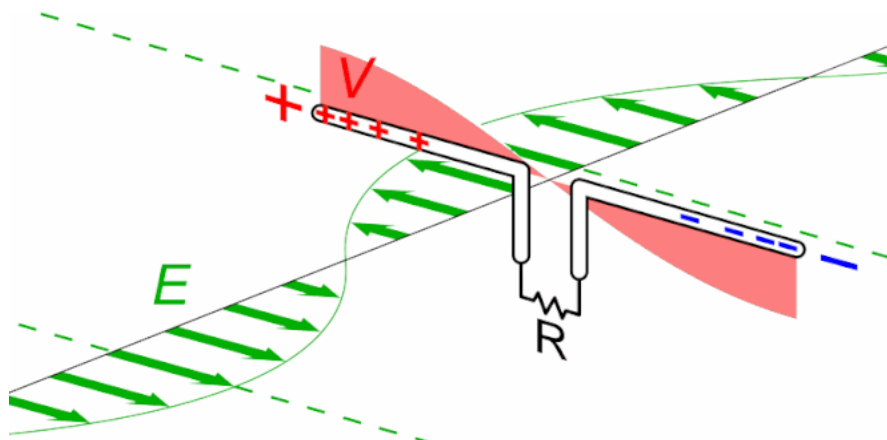
q

out the
position of Q can reach q
faster than the speed of light

Answer: Magnetism

Accelerating Charges: Radiation



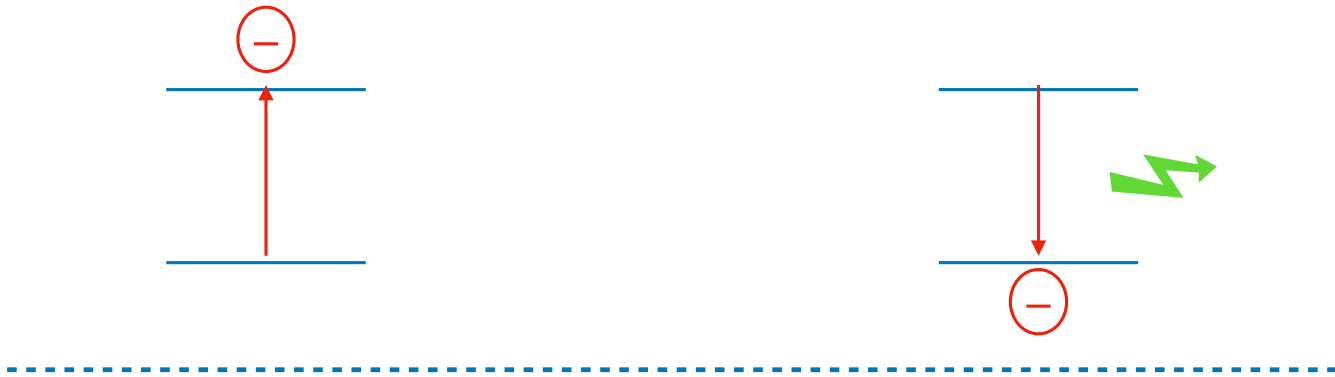


Black Body Radiation

Ultraviolet catastrophe



Quantum Optics



MAXWELL EQUATIONS

$$\nabla \times \vec{E} = -\mu \frac{\partial \vec{H}}{\partial t}$$

$$\nabla \times \vec{H} = \epsilon \frac{\partial \vec{E}}{\partial t} + \vec{J}$$

$$\nabla \cdot \epsilon \vec{E} = \rho$$

Electric Field Notation

$$E(x, y, z, t) = |A(x, y, z)| \cos[\omega t - \phi'(x, y, z)]$$

$$E(x, y, z, t) = |A(x, y, z)| \cos[\omega t - kz - \phi(x, y, z)]$$

$$E(x, y, z, t) = \text{Re}\{|A(x, y, z)| e^{-j\phi(x, y, z)} e^{j[\omega t - kz]}\}$$

Plane waves

$$\nabla \times \nabla \times \vec{E} = -\mu \frac{\partial(\nabla \times \vec{H})}{\partial t} = -\epsilon\mu \frac{\partial^2 \vec{E}}{\partial t^2}$$

$$\nabla \times \nabla \times \vec{E} = -\nabla^2 \vec{E} + \nabla(\nabla \cdot \vec{E}) = -\nabla^2 \vec{E}$$

This gives us the wave equation

$$\nabla^2 \vec{E} = \epsilon\mu \frac{\partial^2 \vec{E}}{\partial t^2}$$

This is a vector equation. We can write in terms of each of the elements of the vector, for example E_x .

$$\frac{\partial^2 E_x}{\partial x^2} + \frac{\partial^2 E_x}{\partial y^2} + \frac{\partial^2 E_x}{\partial z^2} = -\omega^2 \mu \epsilon E_x$$

Solution to this is the plane wave:

$$E_x(x, y, z, t) = A \cos(\omega t - k_x x - k_y y - k_z z)$$

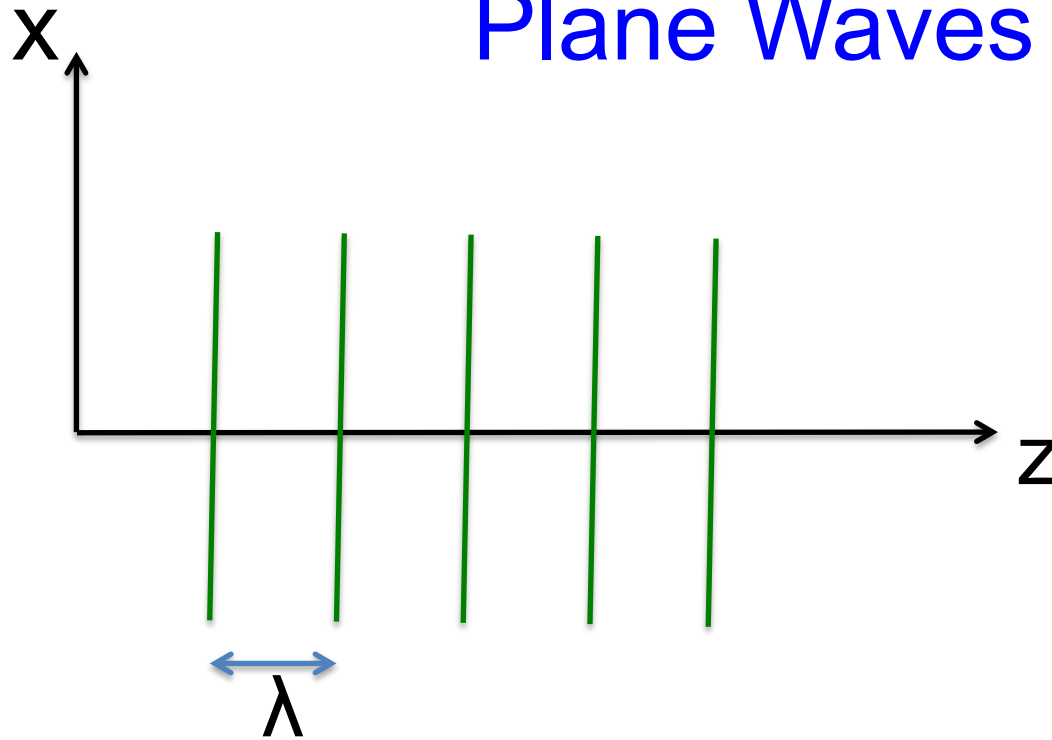
Or in complex notation

$$E_x(x, y, z, t) = A e^{j(\omega t - k_x x - k_y y - k_z z)}$$

Substituting this back into the wave equation

$$k^2 = k_x^2 + k_y^2 + k_z^2 = \omega^2 \mu \epsilon$$

Plane Waves

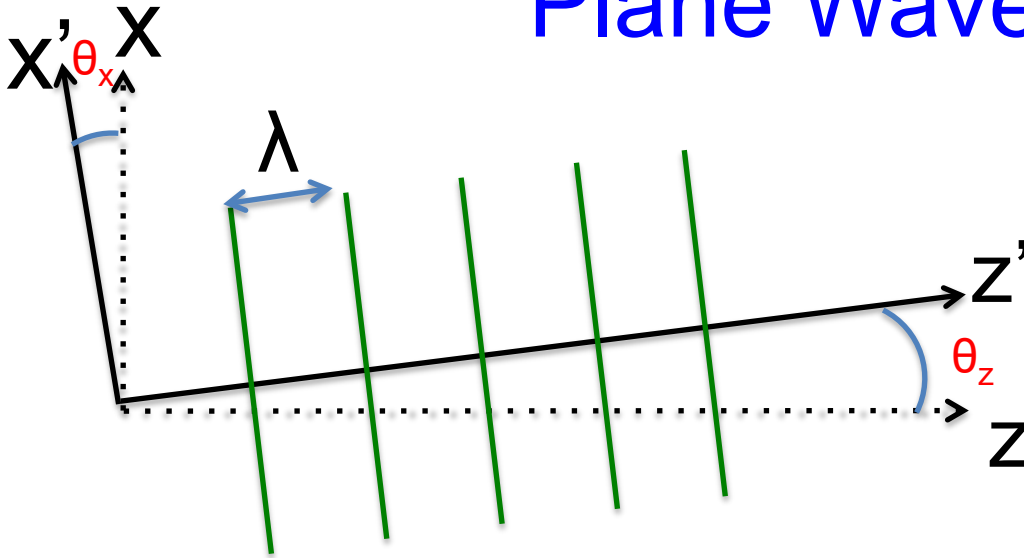


$$\begin{aligned} k &= \frac{2\pi}{\lambda} \\ \omega &= 2\pi\nu \\ c &= \lambda\nu \end{aligned}$$

$$k_x = 0 \Rightarrow E(x, y, z, t) = Ae^{j(\omega t - kz)}$$

$$k = 2\pi / \lambda = \omega / \nu$$

Plane Waves



$$k_x \neq 0 \Rightarrow E(x', y', z', t) = Ae^{j(\omega t - kz')}$$

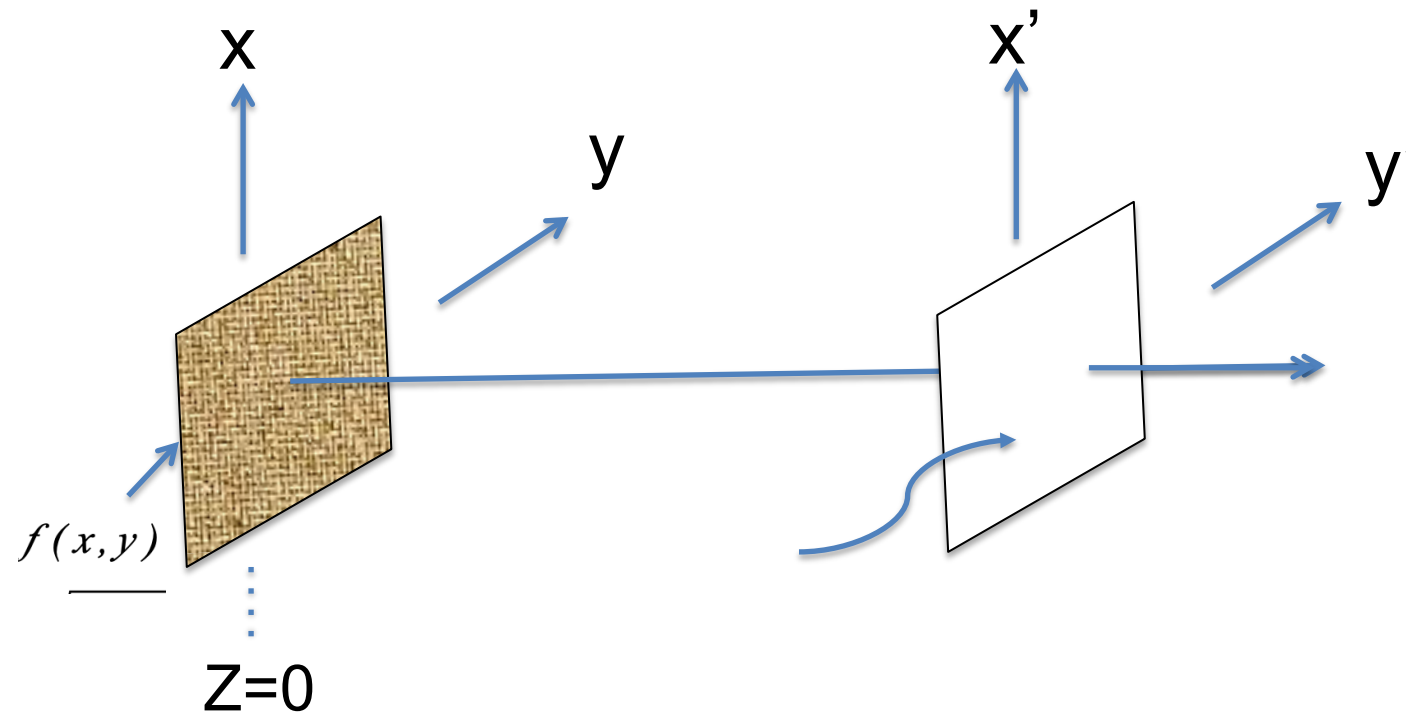
$$z' = x \cos \theta_x + z \cos \theta_z \Rightarrow E = Ae^{j(\omega t - k \cos \theta_x x - k \cos \theta_z z)}$$

$$k_x = k \cos \theta_x \quad k_z = k \cos \theta_z$$

$$k^2 = k_x^2 + k_z^2 = \left(\frac{2\pi}{\lambda}\right)^2 \Rightarrow k_z = \sqrt{k^2 - k_x^2}$$

Eigenmodes and eigenvalues?

Angular spectrum



$$E(x, y, z = 0, t) = Ae^{j\omega t} f(x, y) = Ae^{j\omega t} (2\pi)^2 \iint F(u, v) e^{j2\pi(ux+vy)} du dv$$

$$k_x = 2\pi u \quad k_y = 2\pi v \quad k_z = \sqrt{k^2 - k_x^2 - k_y^2} = k_z = \sqrt{k^2 - (2\pi u)^2 - (2\pi v)^2}$$

$$E(x, y, z \neq 0, t)$$

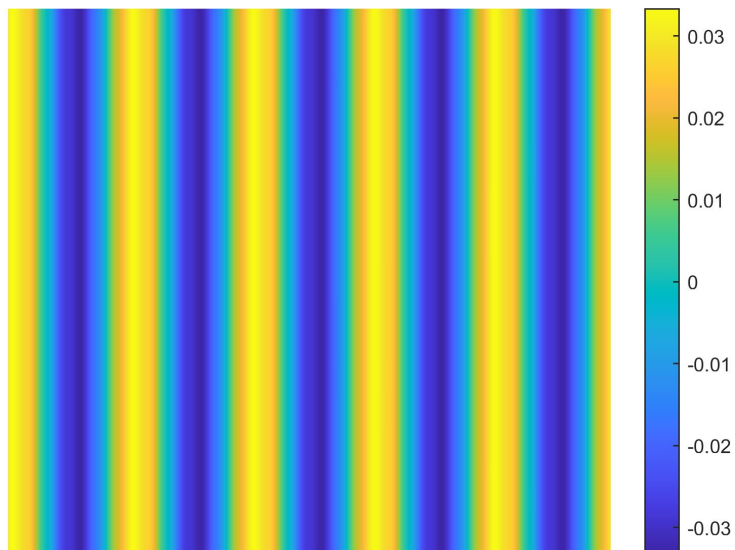
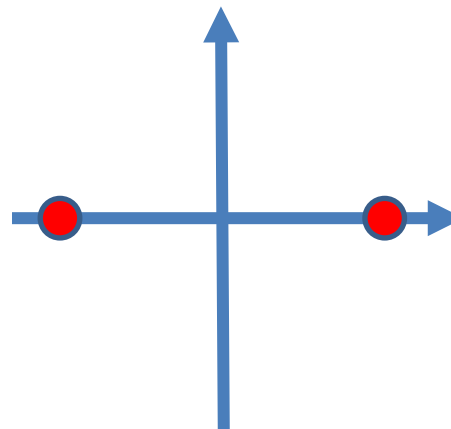
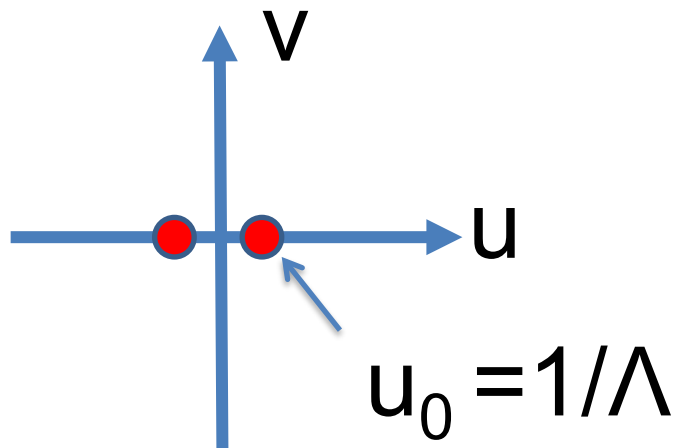
$$= e^{j\omega t} \iint F(u, v) e^{j(2\pi ux + 2\pi vy + \sqrt{k^2 - (2\pi u)^2 - (2\pi v)^2} z)} du dv$$

2D Fourier transform

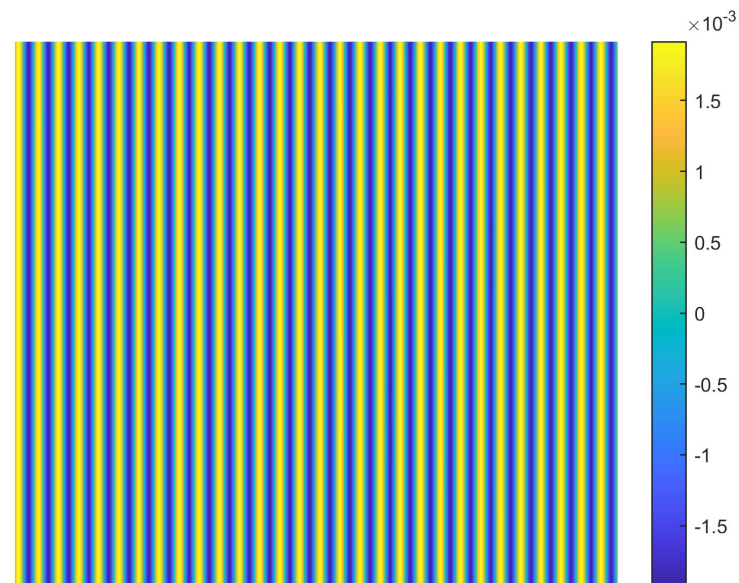
$$F(u, v) = \int f(x, y) e^{-j2\pi(ux+vy)} dx dy$$

$$f(x, y) = \frac{1}{2} + \frac{1}{2} \cos(2\pi u_0 x + 2\pi v_0 y)$$

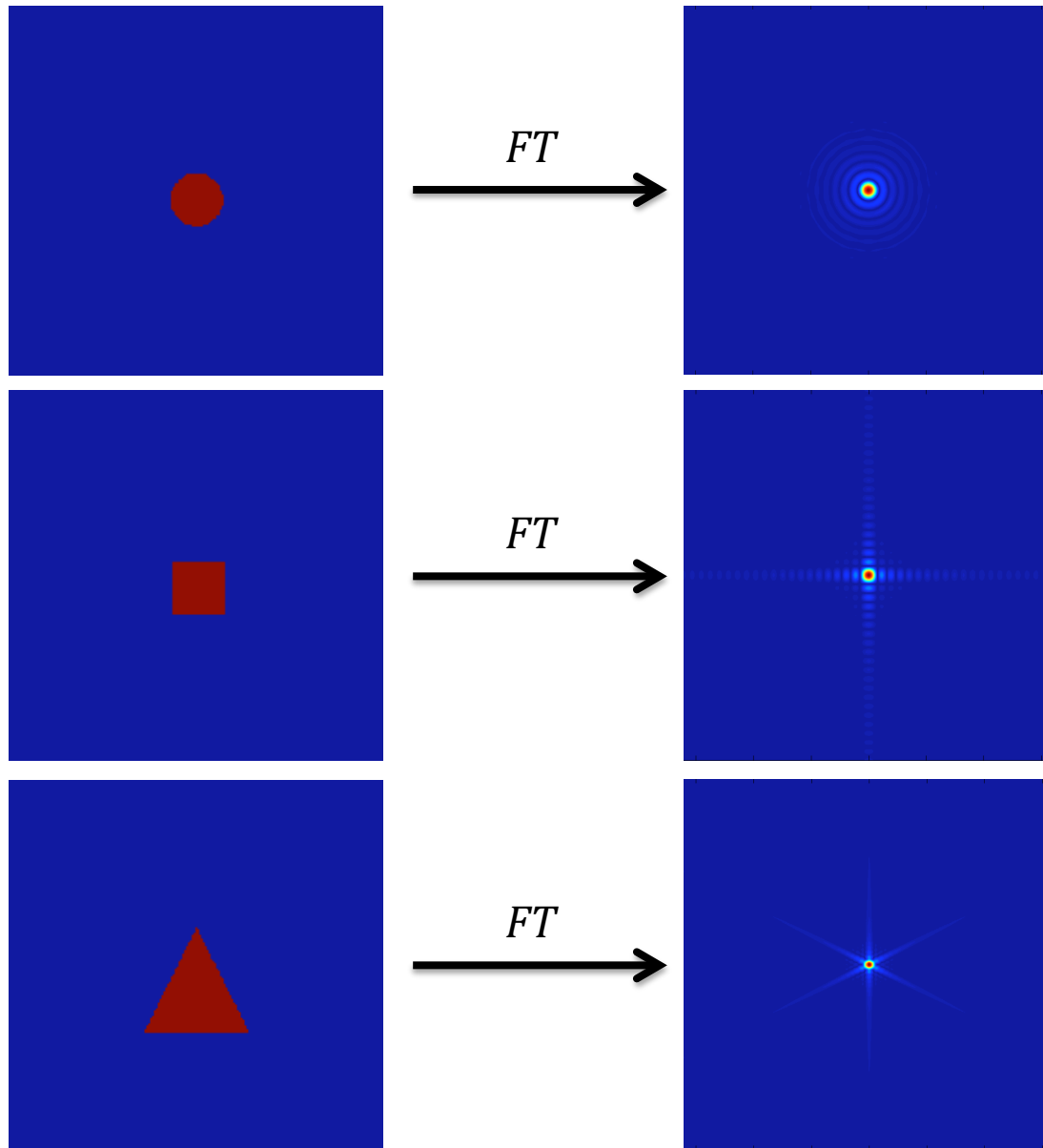
$$\Rightarrow F(u, v) = \frac{1}{2} \delta(u, v) + \frac{1}{4} \delta(u \pm u_0, v \pm v_0)$$



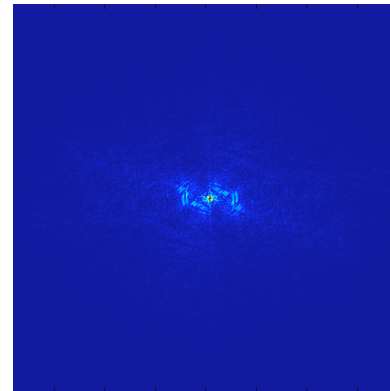
Λ



Fourier Transform pairs



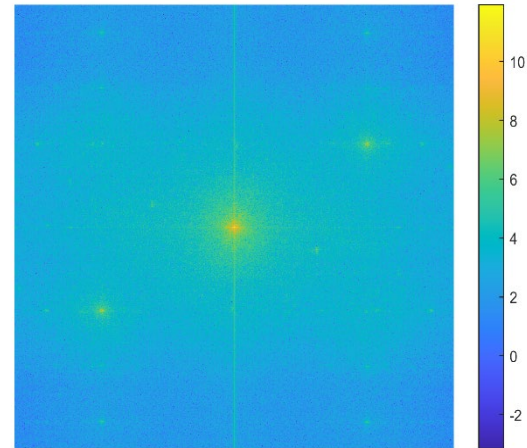
Fourier Transform pair



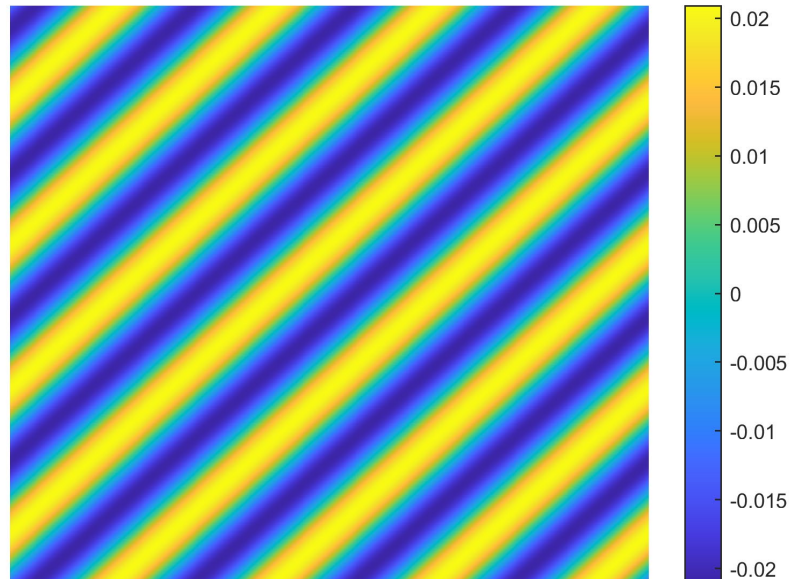
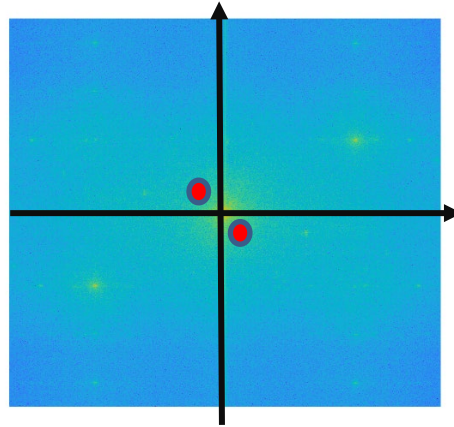
Isaac Newton

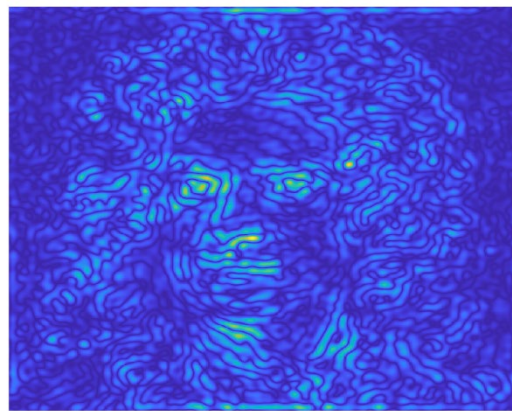
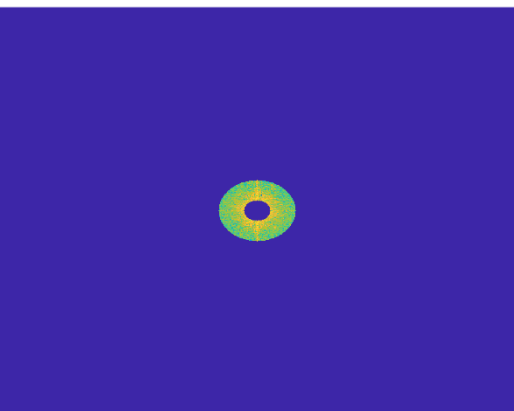
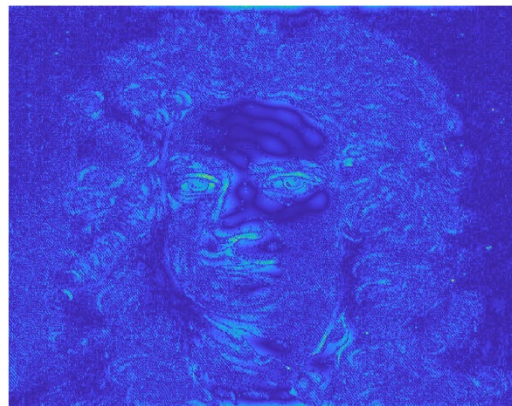
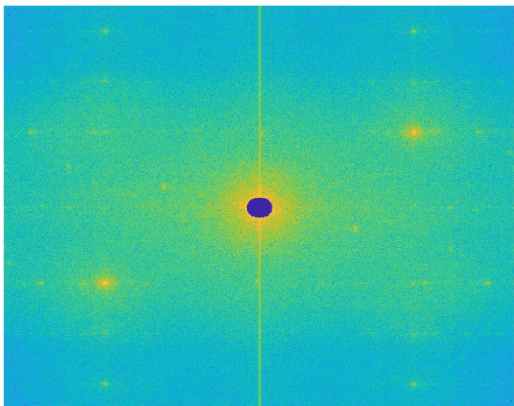
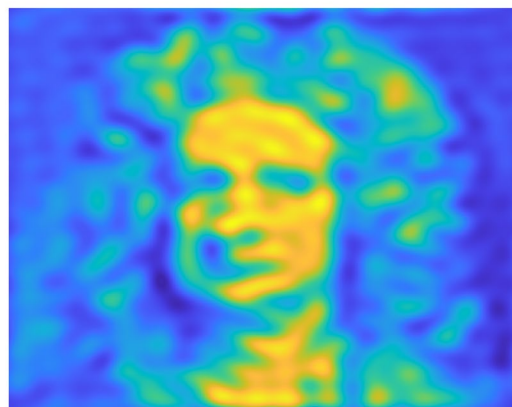
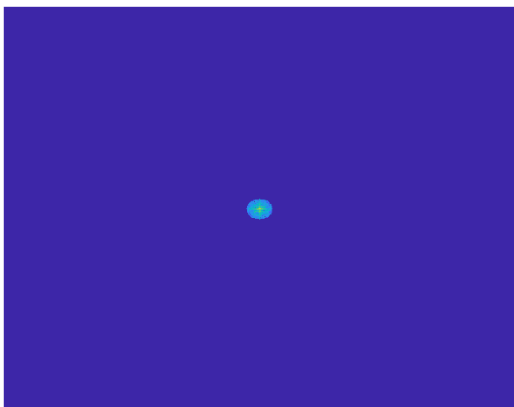


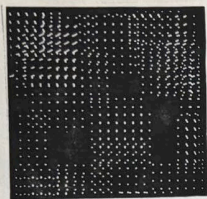
Log of
absolute value of
2D FFT of
Sir Isaac Newton



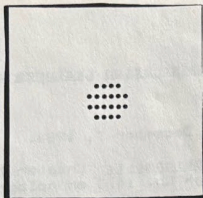
Extraction of a frequency component of Isaac Newton



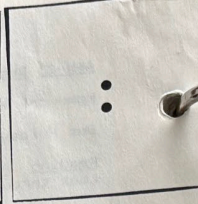


IMAGES

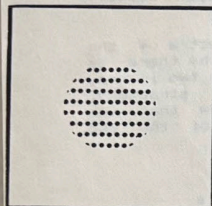
A



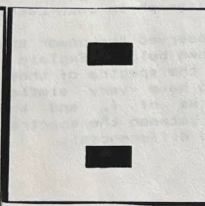
B



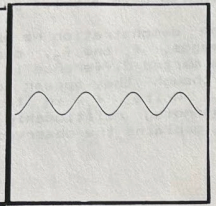
C



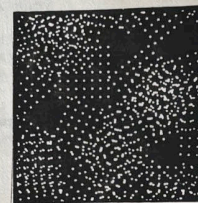
D



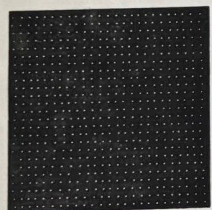
E



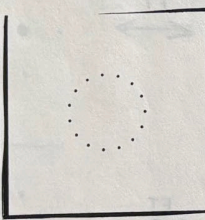
F



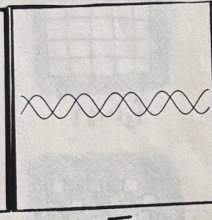
G



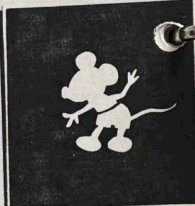
H



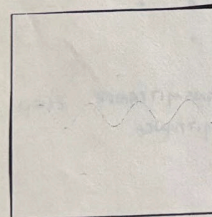
I



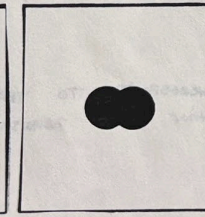
J



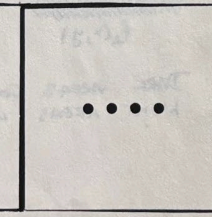
K



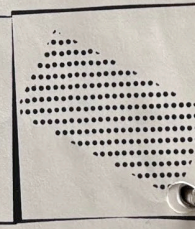
L



M



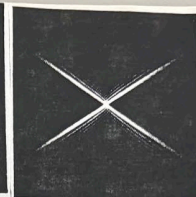
N



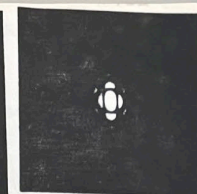
O

FOURIER
TRANSFORMS

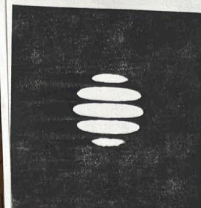
1



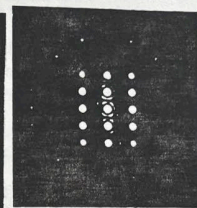
2



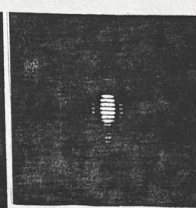
3



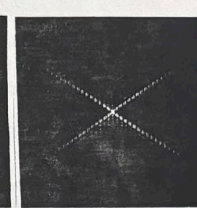
4



5



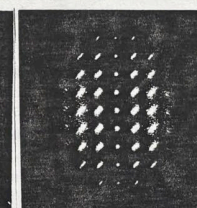
6



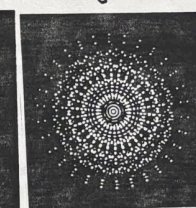
7



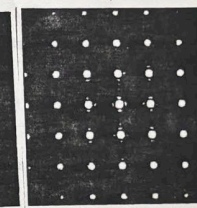
8



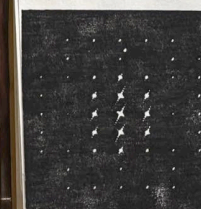
9



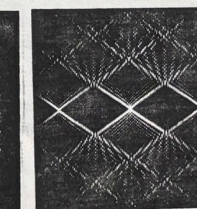
10



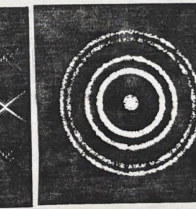
11



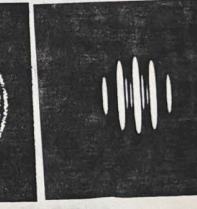
12



13

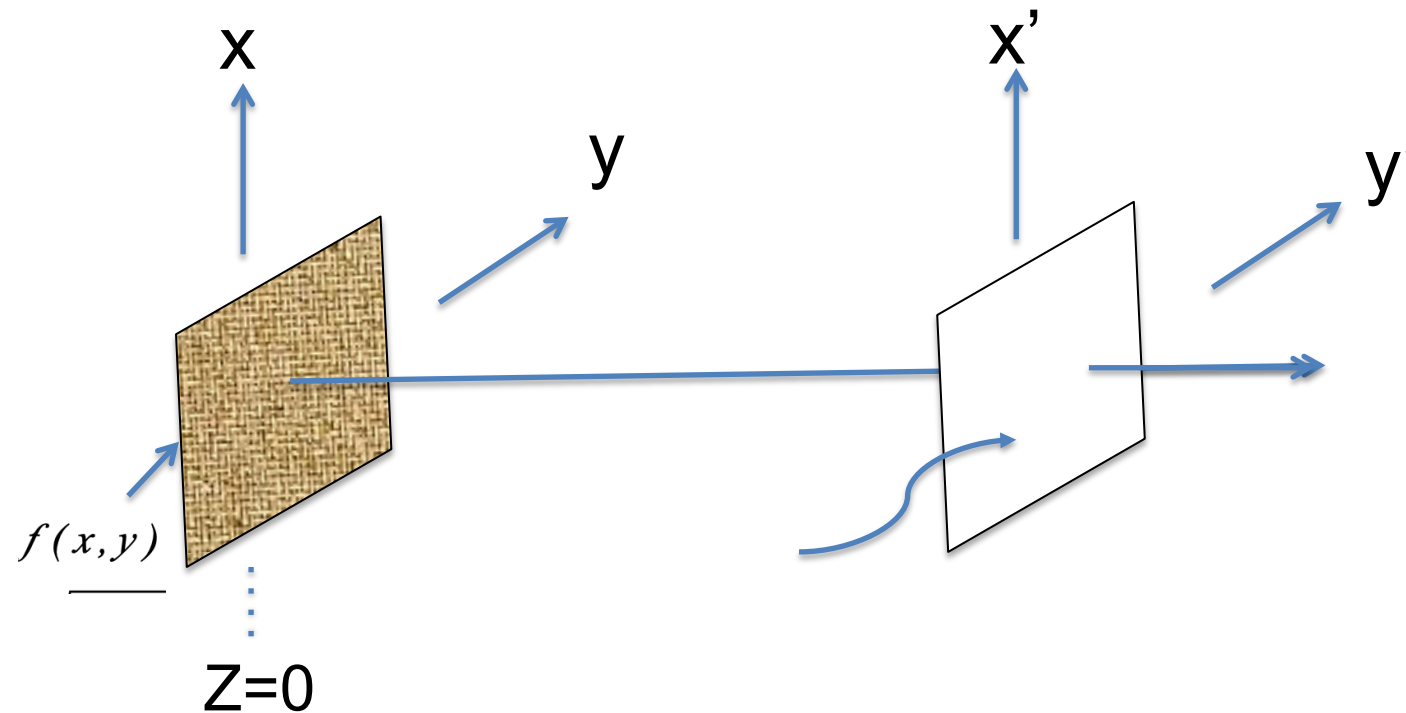


14



15

Angular spectrum



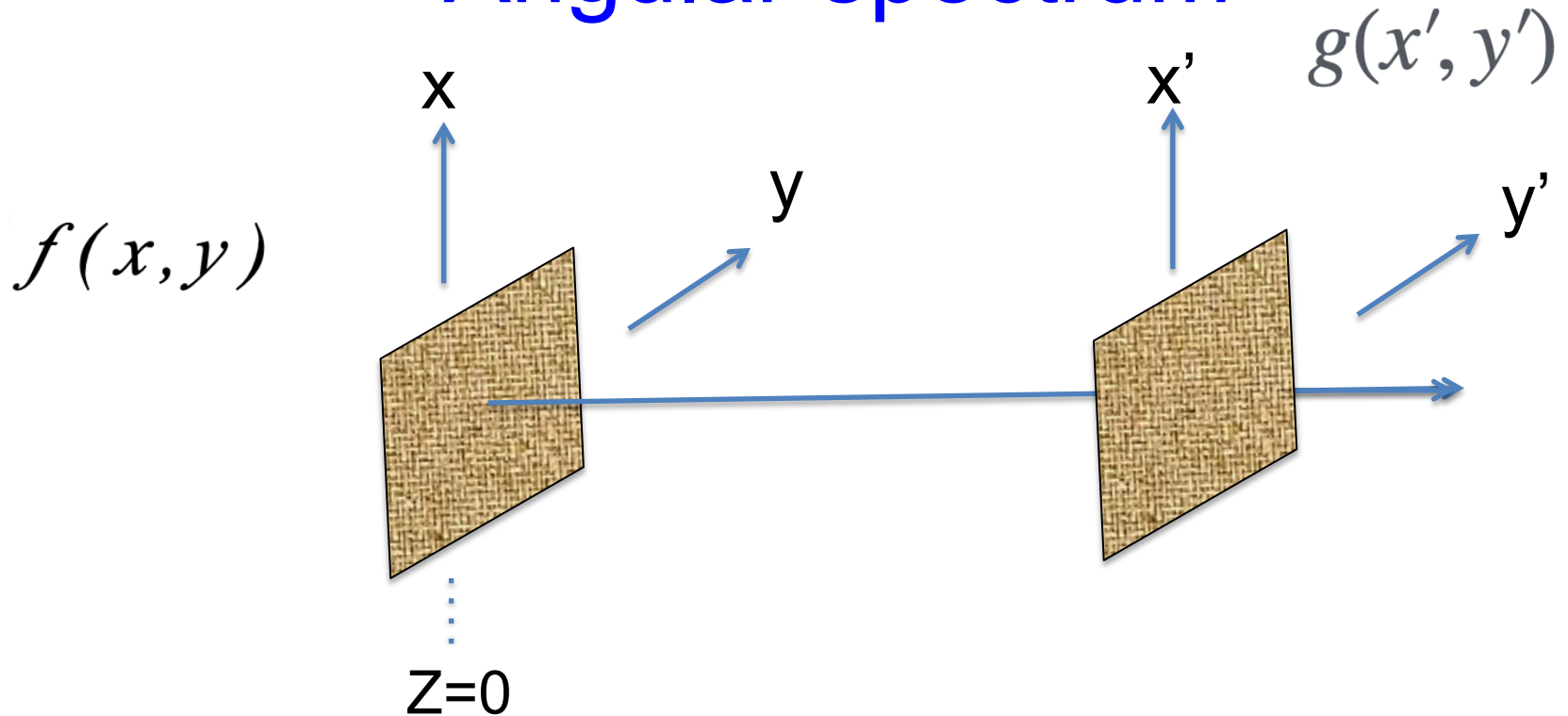
$$E(x, y, z = 0, t) = Ae^{j\omega t} f(x, y) = Ae^{j\omega t} (2\pi)^2 \iint F(u, v) e^{j2\pi(ux+vy)} du dv$$

$$k_x = 2\pi u \quad k_y = 2\pi v \quad k_z = \sqrt{k^2 - k_x^2 - k_y^2} = k_z = \sqrt{k^2 - (2\pi u)^2 - (2\pi v)^2}$$

$$E(x, y, z \neq 0, t)$$

$$= e^{j\omega t} \iint F(u, v) e^{j(2\pi ux + 2\pi vy + \sqrt{k^2 - (2\pi u)^2 - (2\pi v)^2} z)} du dv$$

Angular spectrum



3 steps

1. Take the 2D Fourier transform of $f(x, y)$
2. Multiply each spatial frequency component by its eigenvalue
3. Take the inverse 2D Fourier transform to get $g(x', y')$

Angular Spectrum Method

Step 1
2D FT

$$F(u, v) = \int f(x, y) e^{-2j\pi(ux+vy)} dx dy$$

Step 2
Propagate
each
wave

$$G(u, v) = F(u, v) e^{j\sqrt{k^2 - (2\pi u)^2 - (2\pi v)^2} z}$$

Step 3
Inverse FT

$$g(x', y') = \int G(u, v) e^{j2\pi(xu+vy)} du dv$$

ChatGPT

please give me a python code for propagation of light in free space using the angular spectrum

Here is a Python script to simulate the propagation of light in free space using the angular spectrum method. This script uses `numpy` and `matplotlib` for calculations and visualization, and `scipy.fft` for Fourier transforms.

Angular Spectrum

Answer in chat instead >

Angular Spectrum

Run

```
1 import numpy as np
2 import matplotlib.pyplot as plt
3 from scipy.fftpack import fft2, ifft2, fftshift, ifftshift
4
5 def angular_spectrum_propagation(U0, wavelength, dx, dy, z):
6     """
7     Propagates a wavefield U0 over a distance z using the angular spectrum method.
8     :param U0: Initial field distribution (2D numpy array)
9     :param wavelength: Wavelength of the wave (in meters)
10    :param dx: Sampling interval in x (meters)
11    :param dy: Sampling interval in y (meters)
12    :param z: Propagation distance (meters)
13    :return: Propagated field U_z
14    """
15    k = 2 * np.pi / wavelength # Wavenumber
16    ny, nx = U0.shape # Get the size of the input field
17
18    # Spatial frequency coordinates
19    fx = np.fft.fftfreq(nx, dx)
20    fy = np.fft.fftfreq(ny, dy)
21    FX, FY = np.meshgrid(fx, fy)
22
23    # Compute the transfer function (angular spectrum propagation)
24    H = np.exp(1j * z * np.sqrt(k**2 - (2 * np.pi * FX)**2 - (2 * np.pi * FY)**2))
25    H[np.isnan(H)] = 0 # Avoid NaN values for evanescent waves
26
27    # Compute the angular spectrum
28    U0_spectrum = fft2(U0)
29    U_z_spectrum = U0_spectrum * H # Multiply in Fourier space
30    U_z = ifft2(U_z_spectrum) # Inverse Fourier transform
31
```

The 3 steps

ChatGPT ▾

please give me a python code for propagation of light in free space using the angular spectrum

Here is a Python script to simulate the propagation of light in free space using the angular spectrum method. This script uses `numpy` and `matplotlib` for calculations and visualization, and `scipy.fft` for Fourier transforms.

Angular Spectrum

Answer in chat instead >

Here is a Python script that simulates the propagation of light in free space using the

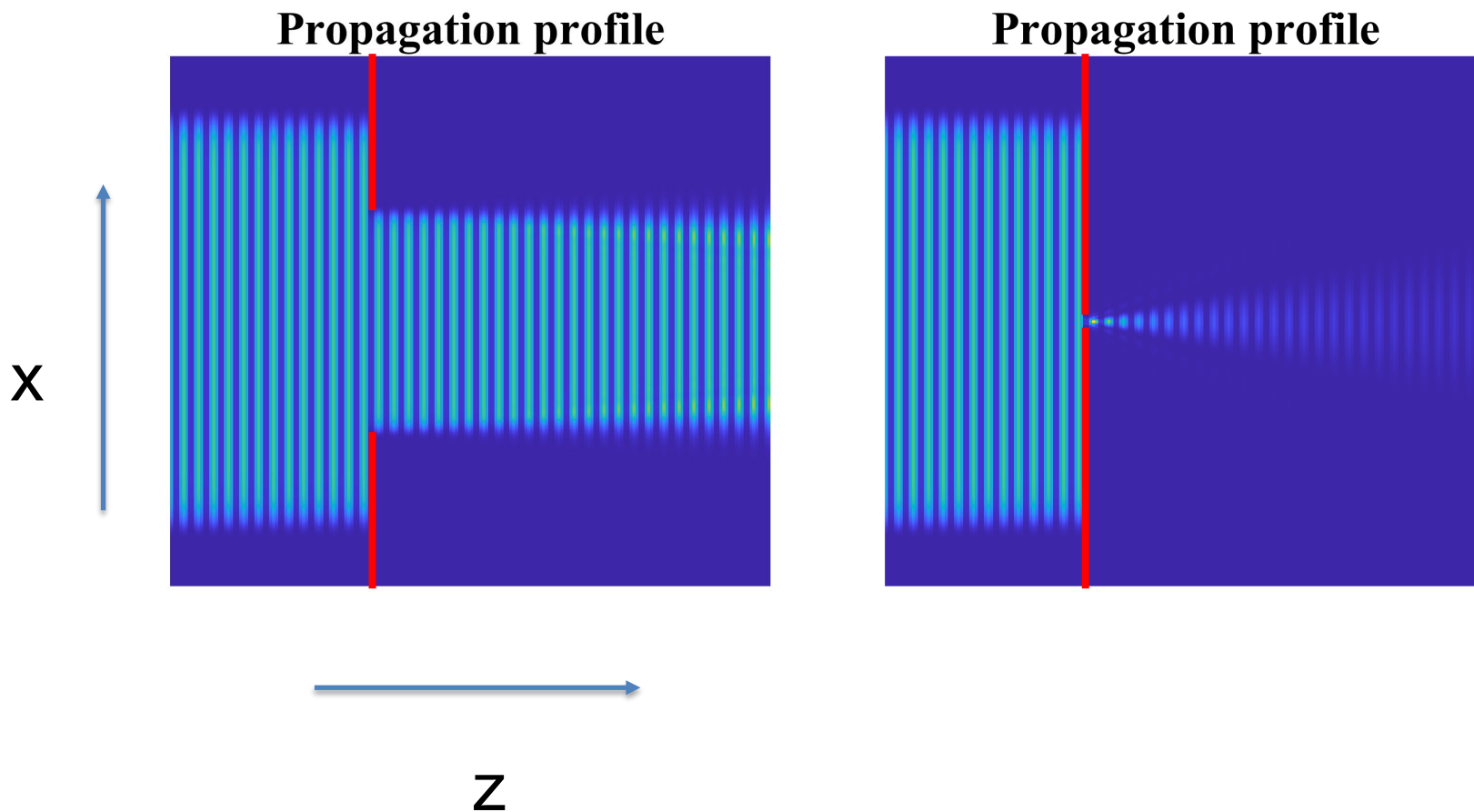
Angular Spectrum ▾

🔄 ⏪ ⏩ 📄 ⬆️ ⬆️ ⬆️ Run

```
31
32     return U_z
33
34     # Define parameters
35     wavelength = 500e-9 # 500 nm (visible light)
36     dx = dy = 10e-6 # 10 microns pixel size
37     z = 0.01 # 1 cm propagation
38
39     # Create an initial Gaussian beam profile
40     x = np.linspace(-500e-6, 500e-6, 100)
41     y = np.linspace(-500e-6, 500e-6, 100)
42     X, Y = np.meshgrid(x, y)
43     U0 = np.exp(-(X**2 + Y**2) / (100e-6)**2) # Gaussian profile
44
45     # Propagate the field
46     U_z = angular_spectrum_propagation(U0, wavelength, dx, dy, z)
47
48     # Plot results
49     plt.figure(figsize=(10, 5))
50     plt.subplot(1, 2, 1)
51     plt.imshow(np.abs(U0), extent=[x.min(), x.max(), y.min(), y.max()])
52     plt.title("Initial Field |U0|")
53     plt.colorbar()
54
55     plt.subplot(1, 2, 2)
56     plt.imshow(np.abs(U_z), extent=[x.min(), x.max(), y.min(), y.max()])
57     plt.title("Propagated Field |U_z|")
58     plt.colorbar()
59
60     plt.show()
61
```

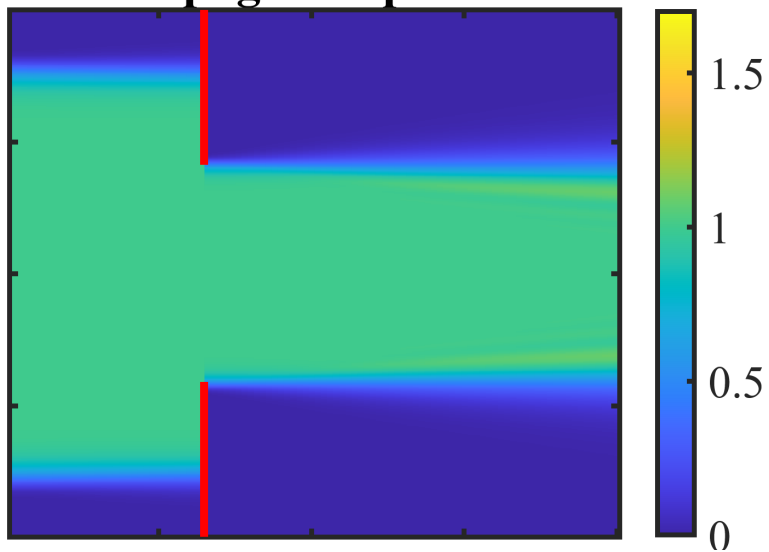
Diffraction of light

x-z plot

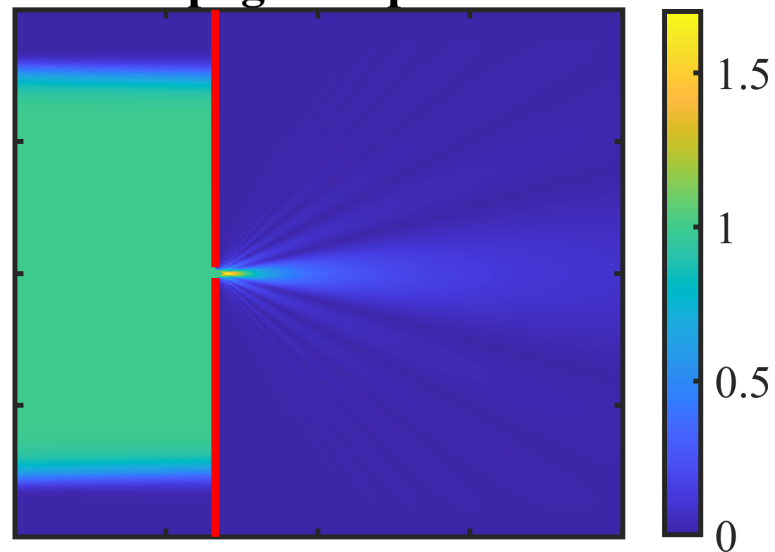


Envelope

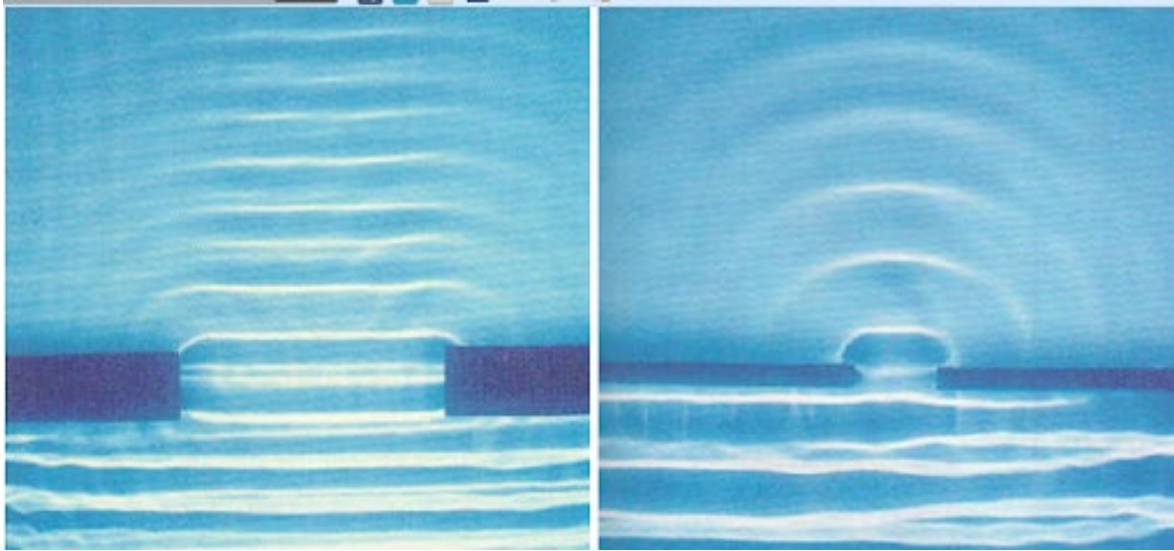
Propagation profile



Propagation profile



Diffraction of water waves



Diffraction of circular aperture

x-z plus xy-z plots



Phase vs amplitude comparison

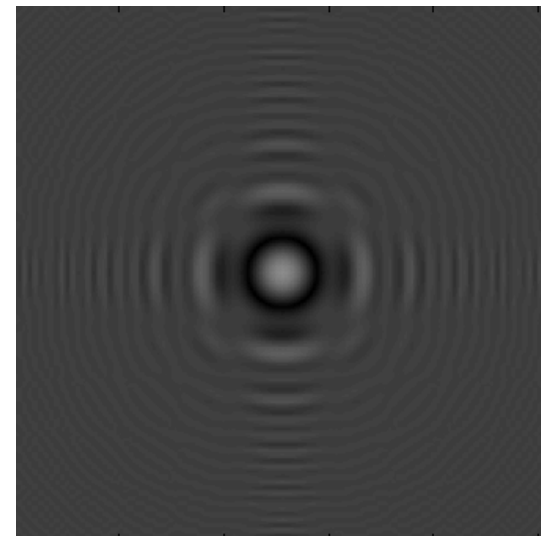
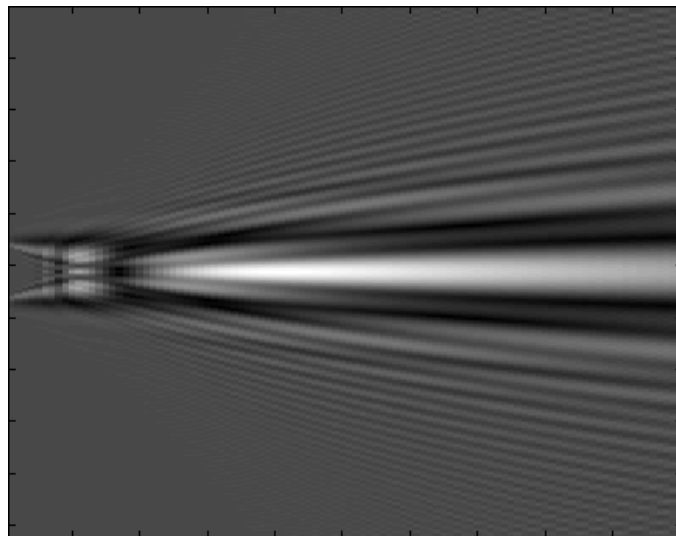
$\lambda = 800\text{nm}$

Input

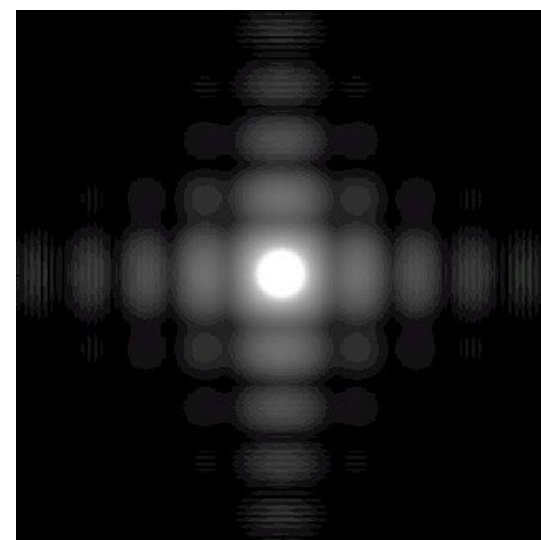
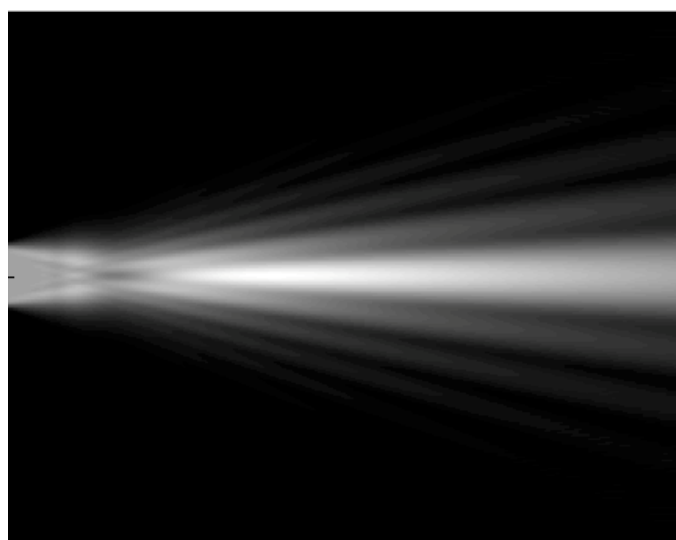
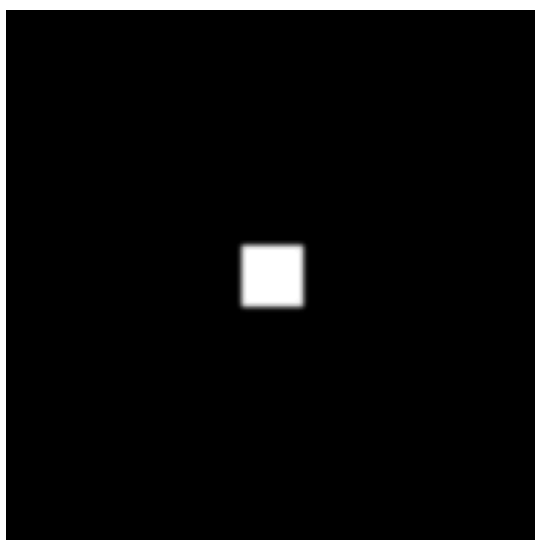
Propagation profile

Output

Phase object

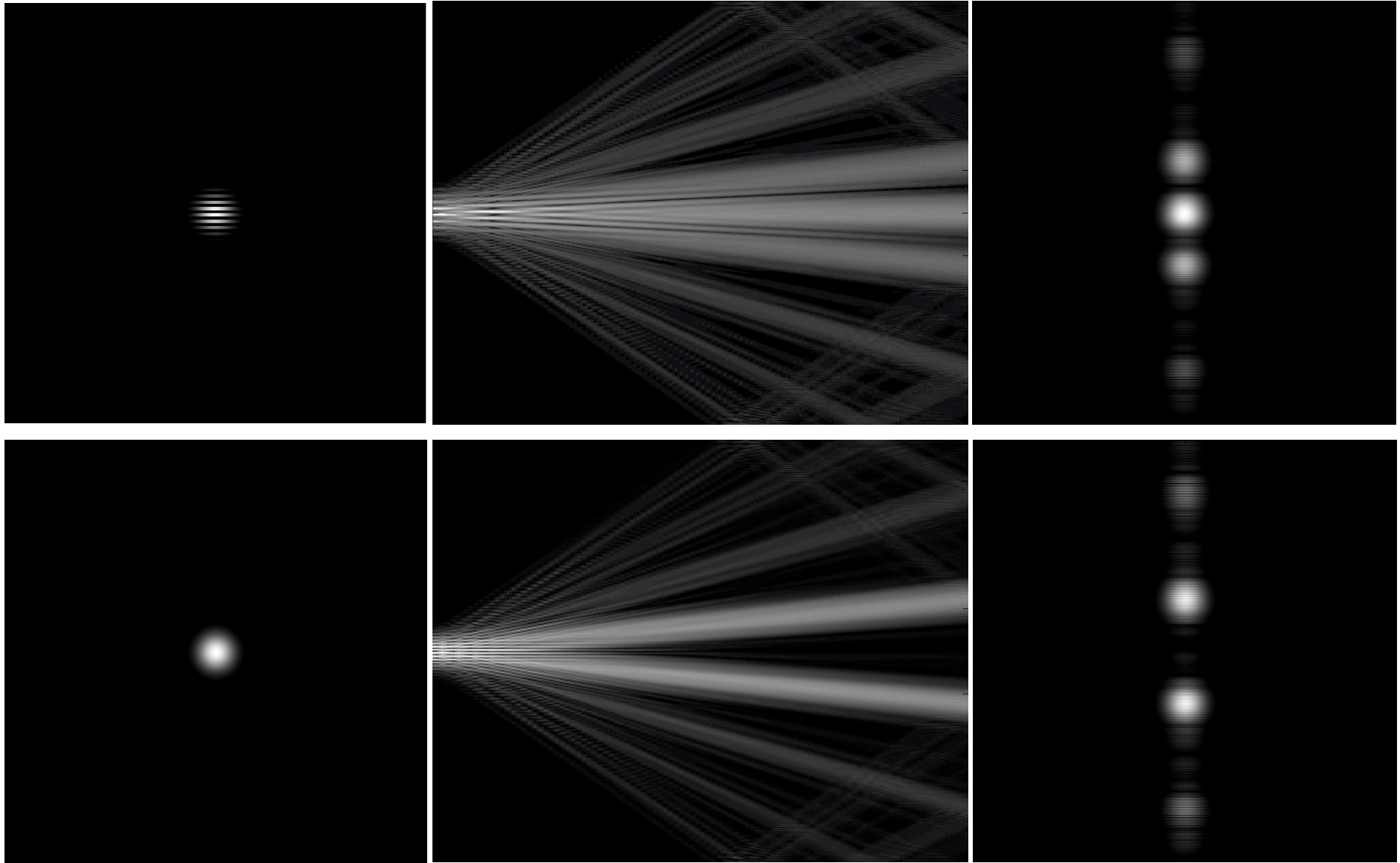


Amplitude object

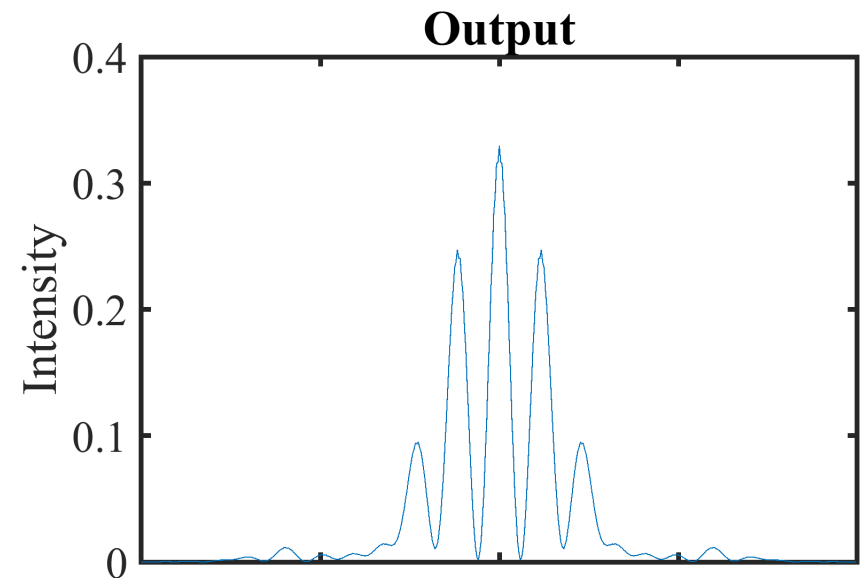
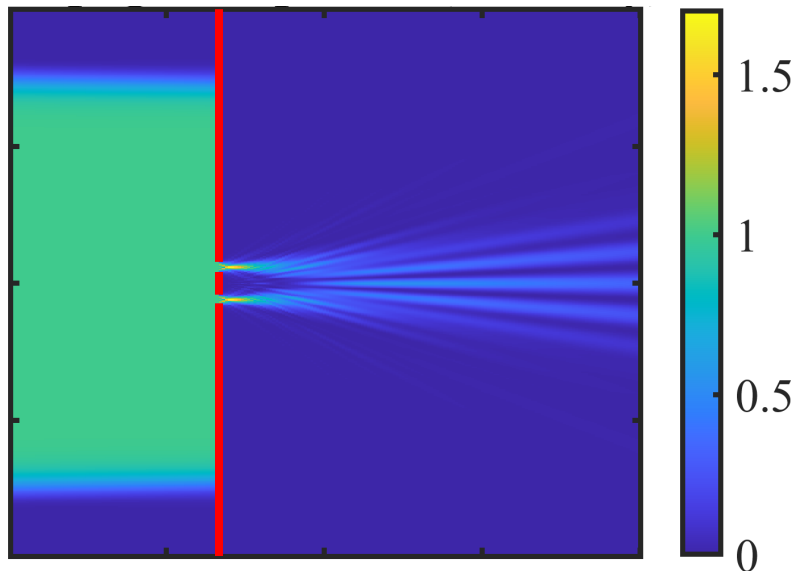


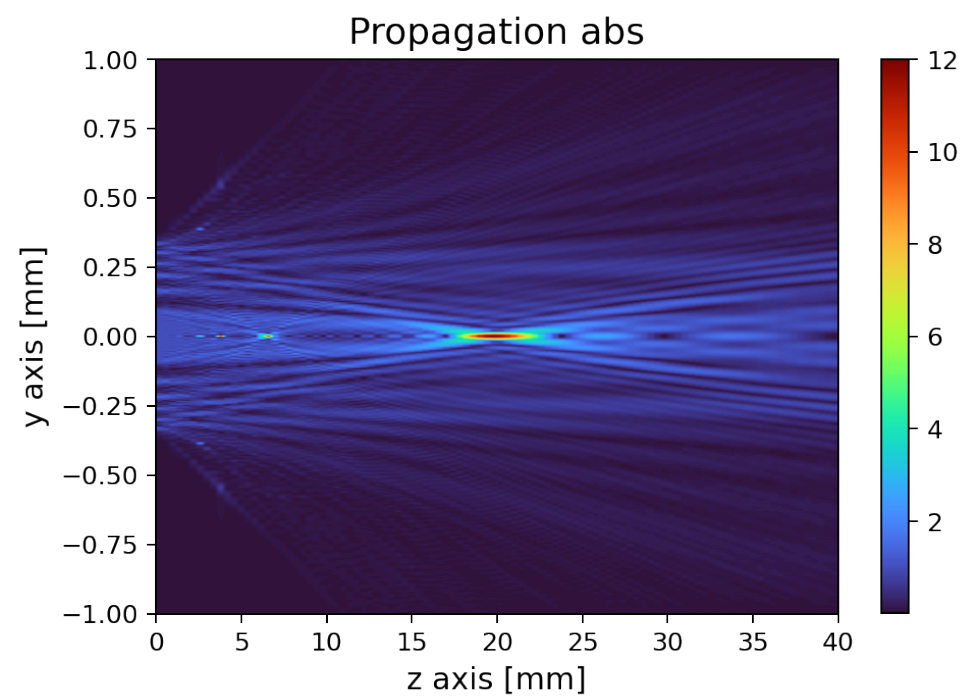
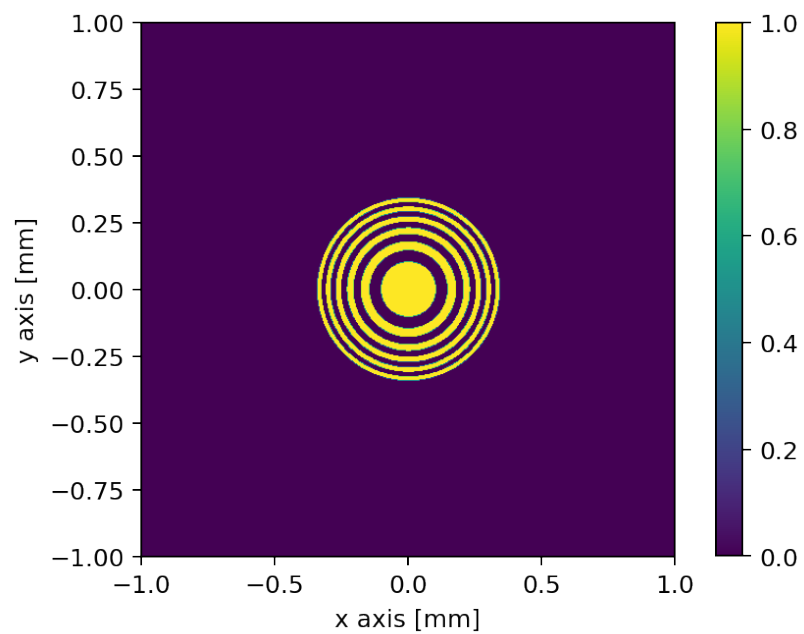
Amplitude and Phase Gratings

$\lambda = 800\text{nm}$



Young's experiment





Diffraction Newton



Diffraction Sir Isaac Newton vs Square aperture



Air Force chart

