pip install numpy matplotlib astropy scipy

import numpy as np

import matplotlib.pyplot as plt

from astropy.io import fits

from scipy.fft import fft2, ifft2, fftshift, ifftshift

# Load the convergence map (κ) from FITS file

filename = "hlsp\_frontier\_model\_abell370\_cats\_v4\_kappa.fits"

hdul = fits.open(filename)

kappa = hdul[0].data

hdul.close()

# Fourier space grid

ny, nx = kappa.shape

lx = np.fft.fftfreq(nx).reshape(1, nx) # normalized freq x

ly = np.fft.fftfreq(ny).reshape(ny, 1) # normalized freq y

# Wavevector squared

k\_squared = (lx\*\*2 + ly\*\*2)

k\_squared[0,0] = 1 # avoid divide by zero for DC term

# Fourier transform of convergence

kappa\_ft = fft2(kappa)

# Compute deflection angle components in Fourier space:

# alpha\_hat\_x = -2i kx / k^2 \* kappa\_hat

# alpha\_hat\_y = -2i ky / k^2 \* kappa\_hat

factor = -2j / k\_squared

alpha\_x\_ft = factor \* lx \* kappa\_ft

alpha\_y\_ft = factor \* ly \* kappa\_ft

# Transform back to real space deflection angles

alpha\_x = np.real(ifft2(alpha\_x\_ft))

alpha\_y = np.real(ifft2(alpha\_y\_ft))

# Visualize deflection angles

plt.figure(figsize=(10,10))

plt.quiver(alpha\_x[::20, ::20], alpha\_y[::20, ::20])

plt.title("Deflection Angles from Convergence Map (Sampled every 20 pixels)")

plt.xlabel("Pixel X")

plt.ylabel("Pixel Y")

plt.show()