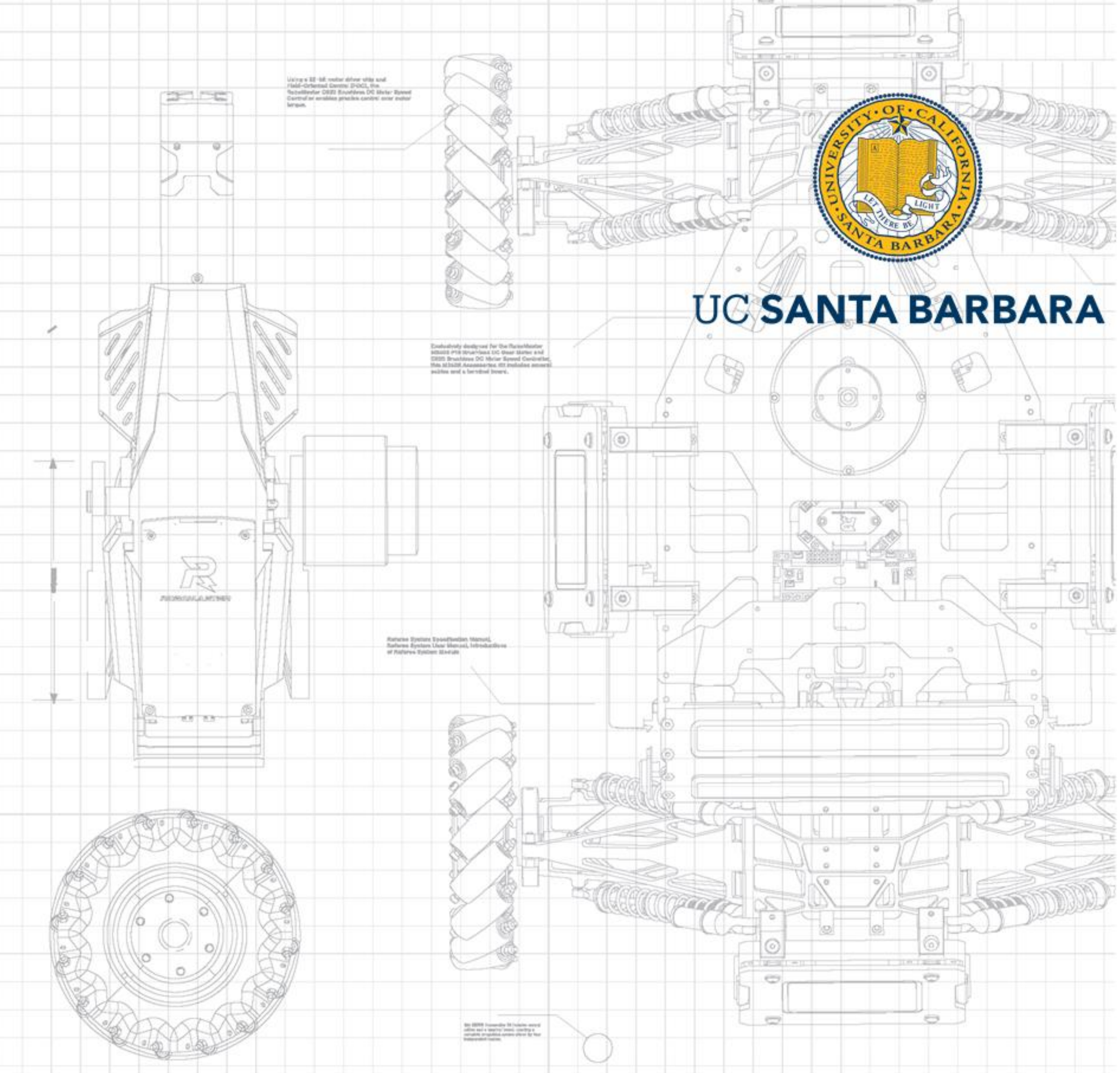

UC SANTA BARBARA

ECE 278C Assignment 1

Peicheng Wu NetID：X311088

2024.01.18

Catalog

1. Platform: VScode

Visual Studio Code, often referred to as VSCode, is a free and open-source code editor developed by Microsoft. It's renowned for its lightweight design, speed, and robust capabilities. Available for Windows, macOS, and Linux, VSCode offers built-in Git integration, an integrated terminal, and a debugger. Its power lies in its extensibility, with a vast marketplace of extensions that add support for various programming languages, debuggers, and tools. The editor's appearance and behavior are highly customizable, and its Intellisense feature provides smart code completions. Thanks to its open-source nature, it has a vibrant community that continually contributes to its development and enhancement.

2. Complex Wave Function in 2 Dimension

$$j\lambda r^{-0.5} * e^{j*2\pi*r*\lambda}$$

3. Question1:

$$\Delta x = \Delta y = \lambda/4$$

$$radius = 30\lambda_0$$

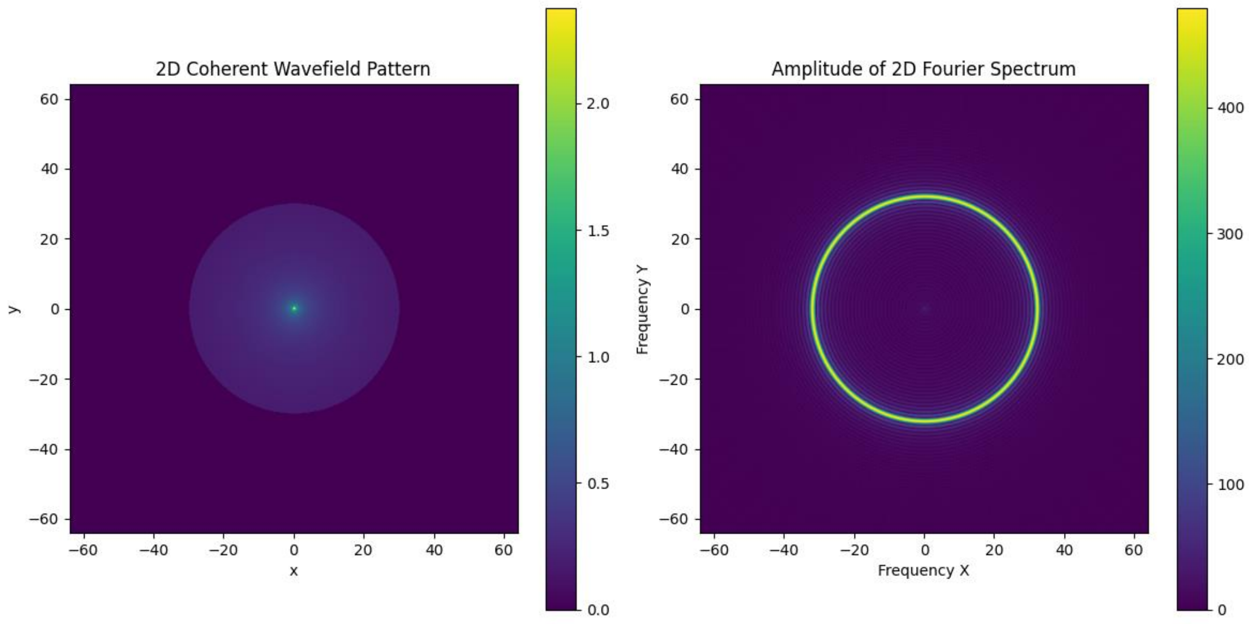

Figure.1 question1

We can see the circle on the left image because the limitation is 30 times lambda, the origin is 1, because the function shows the wave will decrease through transforming, so we can see the origin is the lightest. Also, because the sampling space is lambda/4, so the boundary is from -2/lambda to +2/lambda. The ring's radius is 1/lambda.

4. Question2

	scatters	scatter locations
1	(x_1, y_1)	$(0, +15\lambda_o)$
2	(x_2, y_2)	$(-12\lambda_o, -9\lambda_o)$
3	(x_3, y_3)	$(+12\lambda_o, -9\lambda_o)$

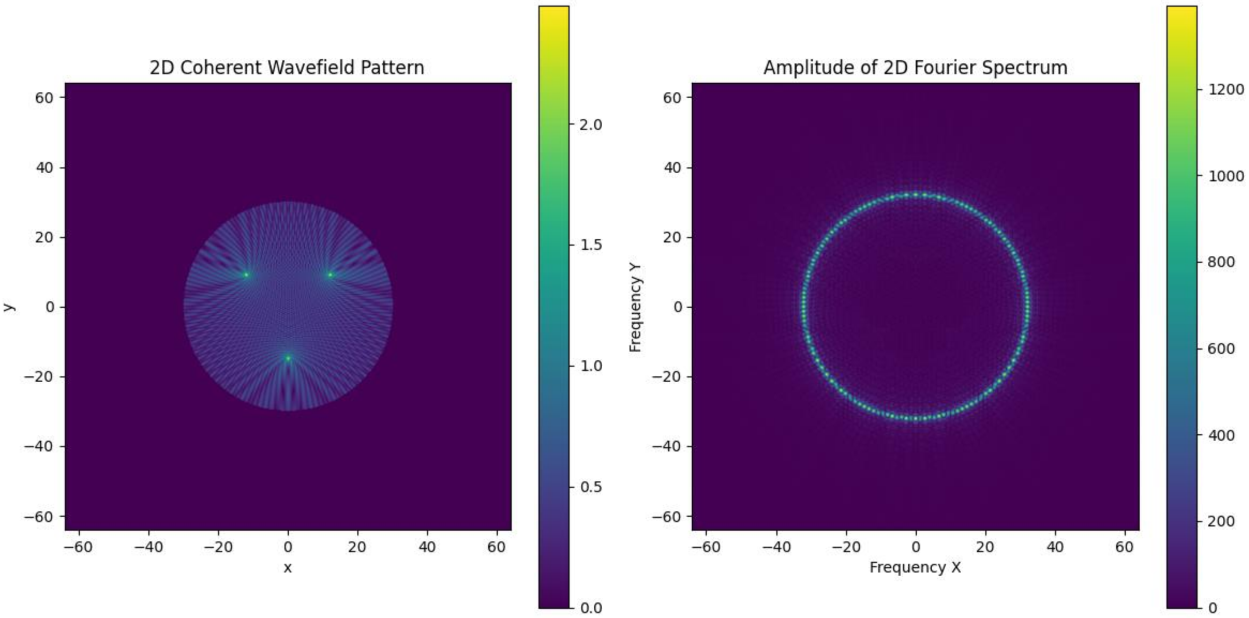

Figure.2 question2

In this part we can see there are three sources which send same waves. So these waves have one f, which means in the f domain we can just see one ring instead of three. Also, different direction has different intensity in f domain.

5. Question3

	scatters	scatter locations	wavelengths
1	(x_1, y_1)	$(0, +15\lambda_o)$	λ_o
2	(x_2, y_2)	$(-12\lambda_o, -9\lambda_o)$	$\lambda_o/2$
3	(x_3, y_3)	$(+12\lambda_o, -9\lambda_o)$	$2\lambda_o$

In this part, we have three different waves because their lambda is different, so we can see the three rings in the f domain. And their size is 1/lambda, 2/lambda and 1/2lambda. Actually we can see other small rings because the limitation is 30 times lambda, it will introduce other f in the f domain.

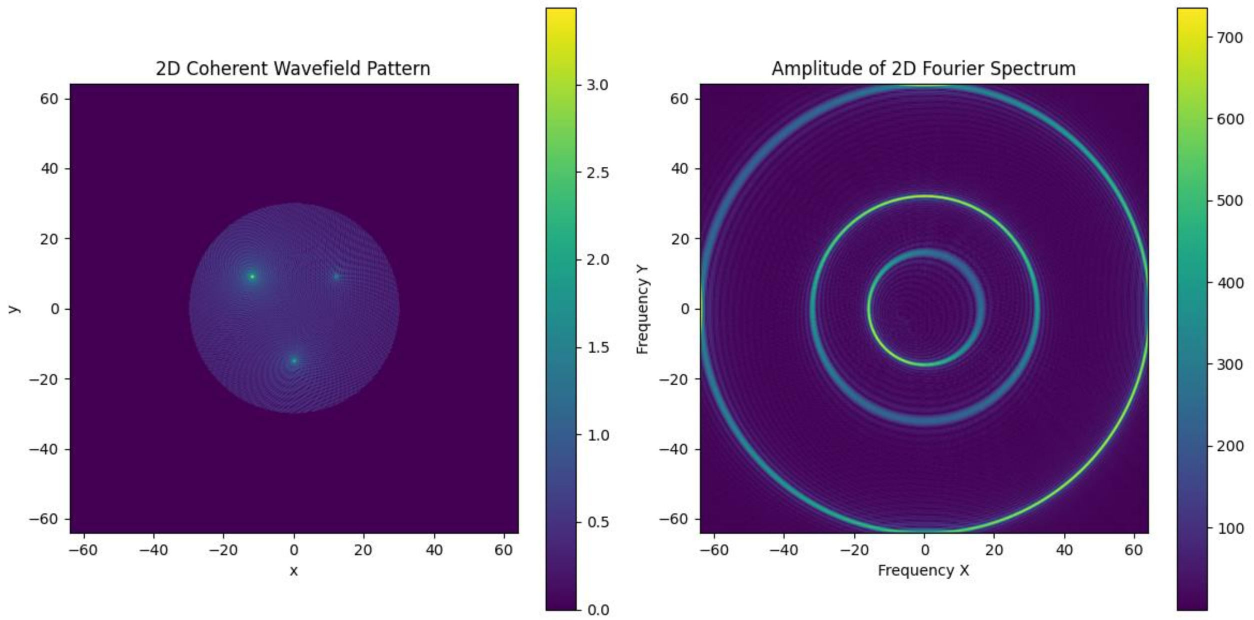

Figure.3 question3

6. Question4

Change the wave function to the Green's Function:

$$f(x) = Ae^{2\pi jr/\lambda}$$

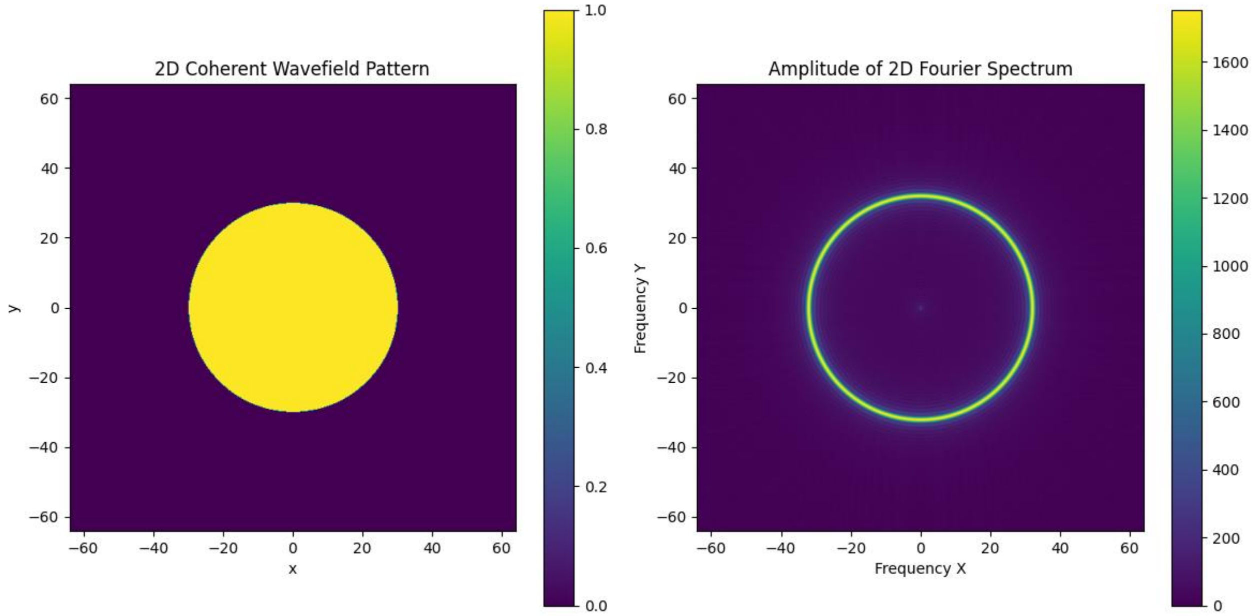

Figure.4 question4

Because the wave will not decrease, so its intensity is same in the space domain. As we can see the yellow circle shows the intensity is one in whole domain. However, it do not change the shape in the f domain(it is still a ring but intensity changed)

7. Question5

Now we can change the sampling space:

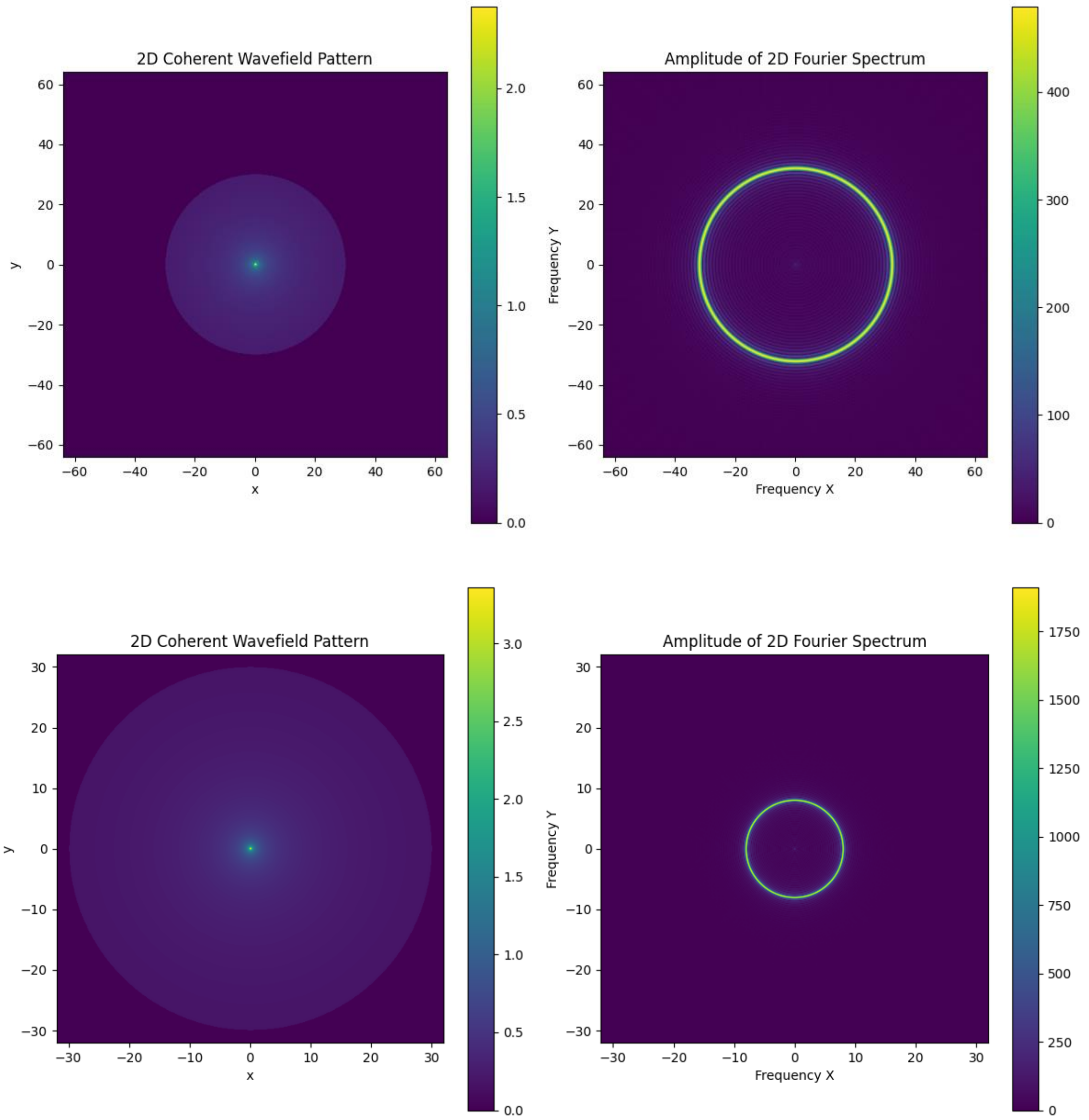

Figure.5 question5

The first one is lambda/4, the second one is lambda/8. In the space domain it do not change a lot, just the size, because the num of points is fixed (512*512). So the distance decreases, the range of space domain decrease. For the f domain we can also get the range is from -4/lambda to 4/lambda.

8. Question6

In this part, we will change the radius of the limitation:

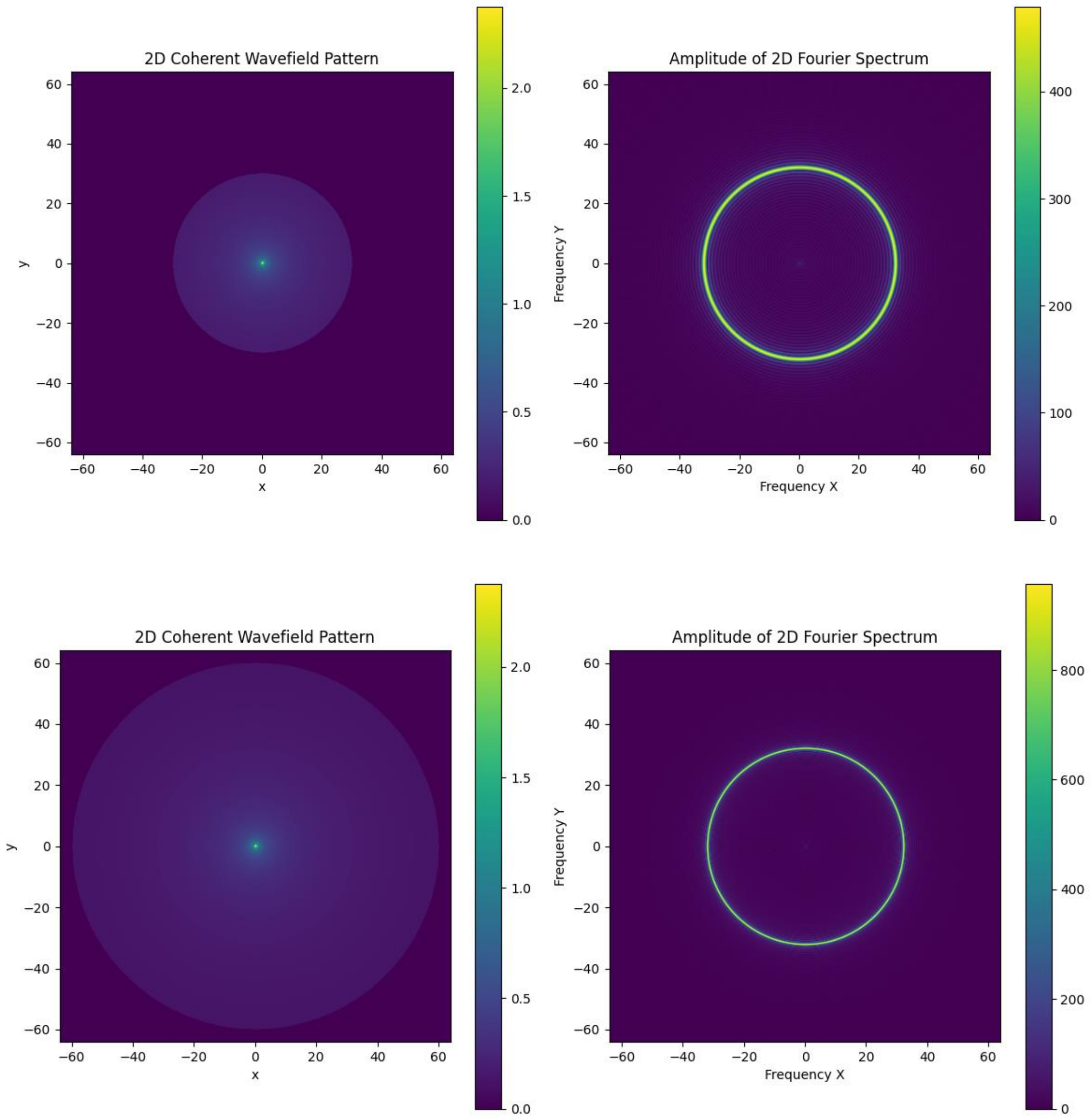

Figure.6 question6

We can see with the increasing of range in space domain, the intensity of ring increase either, but the shape and size does not change because the wave does not change.

9. Appendix

```
import numpy as np
import matplotlib.pyplot as plt
from scipy.fft import fft2, fftshift
import argparse

def question(lambda_0, scatter_locations, num):
    radius_limit = 60 * lambda_0
    N = 512

    delta_x = lambda_0 / 4
    x_max = delta_x * N / 2
    x = np.linspace(-x_max, x_max, N)
    y = np.linspace(-x_max, x_max, N)
    X, Y = np.meshgrid(x, y)
    r = np.sqrt(X**2 + Y**2)

    r[r == 0] = 1
    h = np.zeros((N, N), dtype=complex)
    for lambda_, x_n, y_n in scatter_locations:
        r_n = np.sqrt((X - x_n)**2 + (Y - y_n)**2)
        r_n[r_n == 0] = np.finfo(float).eps
        if num == 4:
            h_n = 1 * np.exp(1j * 2 * np.pi * r_n / lambda_)
            h = h + h_n
        else:
            h_n = (1j * lambda_ * r_n)**(-0.5) * np.exp(1j * 2 * np.pi * r_n / lambda_)
            h = h + h_n

    h[r > radius_limit] = 0

    H = fftshift(fft2(h))

    plt.figure(figsize=(12, 6))
    plt.subplot(1, 2, 1)
    plt.imshow(np.abs(h), extent=(x.min(), x.max(), y.min(), y.max()))
    plt.title("2D Coherent Wavefield Pattern")
    plt.xlabel("x")
    plt.ylabel("y")
    plt.colorbar()

    plt.subplot(1, 2, 2)
    plt.imshow(np.abs(H), extent=(x.min(), x.max(), y.min(), y.max()))
    plt.title("Amplitude of 2D Fourier Spectrum")
    plt.xlabel("Frequency X")
    plt.ylabel("Frequency Y")
    plt.colorbar()
```

```
    plt.tight_layout()
    plt.show()


if __name__ == '__main__':
    parser = argparse.ArgumentParser(description='Different question in the 
assignment')
    parser.add_argument('--Q1', action='store_true', help='Process question 1')
    parser.add_argument('--Q2', action='store_true', help='Process question 2')
    parser.add_argument('--Q3', action='store_true', help='Process question 3')
    parser.add_argument('--Q4', action='store_true', help='Process question 4')

    args = parser.parse_args()

    if args.Q1:
        lambda_0 = 1
        scatter_locations = [(lambda_0, 0, 0)]
        question(lambda_0, scatter_locations, 0)
    elif args.Q2:
        lambda_0 = lambda_1 = lambda_2 = 1
        scatter_locations = [(lambda_0, 0, 15 * lambda_0), (lambda_1, -12 * lambda_0, 
-9 * lambda_0), (lambda_2, 12 * lambda_0, -9 * lambda_0)]
        question(lambda_0, scatter_locations, 0)
    elif args.Q3:
        lambda_0 = 1
        lambda_1 = 0.5
        lambda_2 = 2
        scatter_locations = [(lambda_0, 0, 15 * lambda_0), (lambda_1, -12 * lambda_0, 
-9 * lambda_0), (lambda_2, 12 * lambda_0, -9 * lambda_0)]
        question(lambda_0, scatter_locations, 0)
    elif args.Q4:
        lambda_0 = 1
        scatter_locations = [(lambda_0, 0, 0)]
        question(lambda_0, scatter_locations, 4)
```

You can also find the code on the github: [UCSB-ECE-278C/assignment1 at main · percyance/UCSB-ECE-278C (github.com)]

When you want to run: please type python assignment1.py --Q1 to Q4