Министерство науки и высшего образования Российской Федерации

Лабораторная работа № 1.2.

«Преобразование Фурье. Дифракция Фраунгофера и Френеля. Интерференция. (Часть 2)»

Выполнил: Леко А.А..

Группа: Q4110

Проверила:

Иванова Т. В.

Санкт-Петербург 2023

Задание 1:

Вычислить аналитически и численно распределение интенсивности в картине дифракции в дальней зоне (дифракция Фраунгофера) при помощи двумерного преобразования Фурье (отверстие прямоугольной формы $(\mathrm{rect}(2\mathrm{x})\times\mathrm{rect}(\mathrm{y}))$ в непрозрачном экране).

Аналитическое решение:

$$f(x,y)=rect(x,2y)$$

$$F[f(x,y)]=\frac{\pi}{2}\cdot\mathrm{sinc}(\pi\cdot v_x)\cdot\mathrm{sinc}\left(\frac{\pi}{2}\cdot v_y\right)$$

Вычисленное при помощи python:

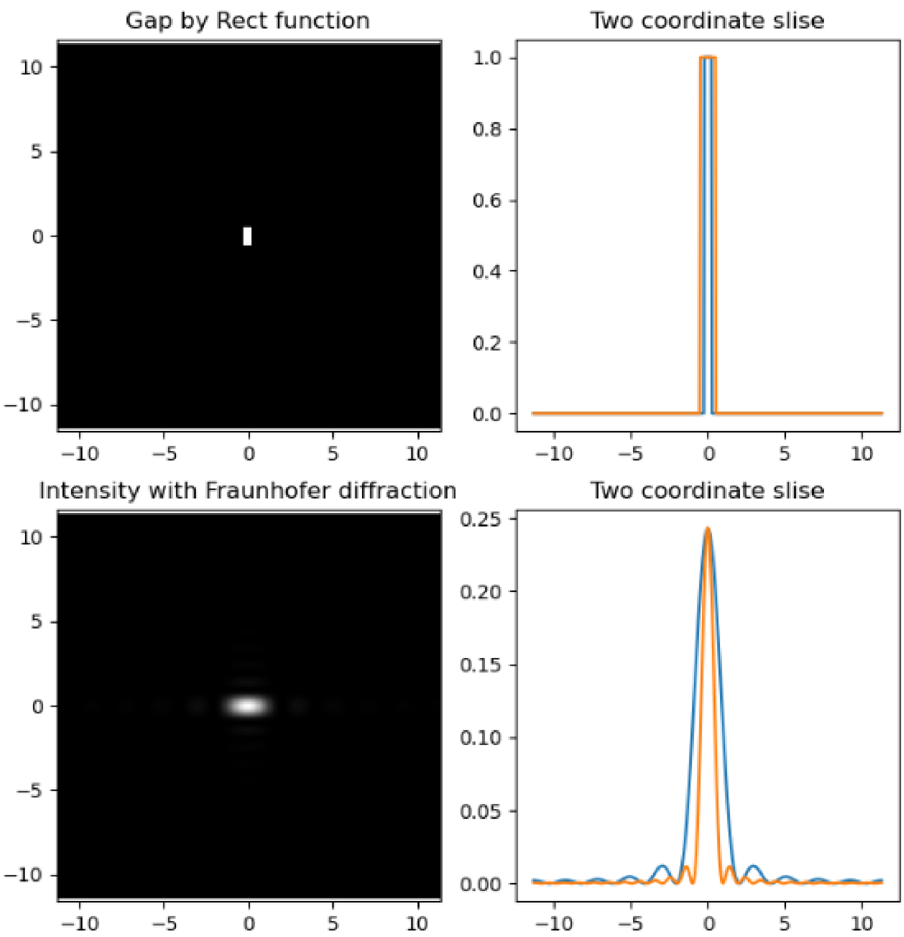

Рисунок 1 – Полутоновые изображения входной функции и распределение интенсивности на экране и их графики в сечениях по осям x и y

Задание 2:

Точечный источник освещает две узкие бесконечно тонкие параллельные щели, расположенные горизонтально на непрозрачном экране (опыт Юнга). Расстояние между щелями 2 мм. Вычислить аналитически и численно распределение интенсивности на плоскости, параллельной экрану и удалённой от него на расстояние 1 м (дифракция Фраунгофера).

Параметры:

$$N = 512$$
$$\Delta x = \sqrt{\frac{1}{N}}$$
$$d = \Delta x$$

Аналитическое решение:

$$f(x,y) = \delta(x-1) + \delta(x+1)$$
$$F[f(x,y)] = \delta(\nu_y)\cdot \exp(2\pi\nu_x) + \delta(\nu_y)\cdot \exp(-2\pi\nu_x) = 2\cdot\delta(\nu_y)\cdot\cos(2\pi\nu_x)$$
$$I = \left|F[f(x,y)]\right|^2 = 4\cdot\left(2\cdot\delta(\nu_y)\cdot\cos(2\pi\nu_x)\right)^2$$

Решение при помощи python:

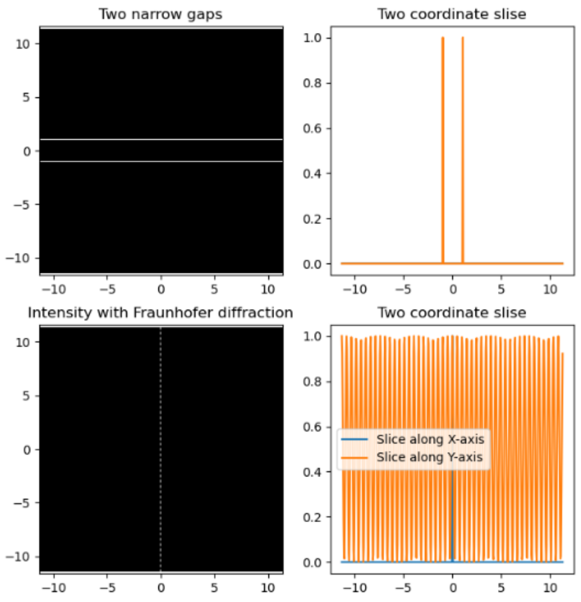

Рисунок 2 – Полутоновые изображения входной функции и распределение интенсивности на экране и их графики в сечениях по осям x и y

Задание 3:

Вычислить численно распределение интенсивности в картине дифракции Френеля на круглом экране.

Входные данные:

$$N = 512$$
$$r = 0.051m$$
$$z = 1000m$$
$$\lambda = 0.5mkm$$
$$n = 1$$
$$x\_\max = 0.256m$$
$$\Delta x = x\_\max \cdot \frac{2}{N}$$

Дифракционная картина полученная в Python:

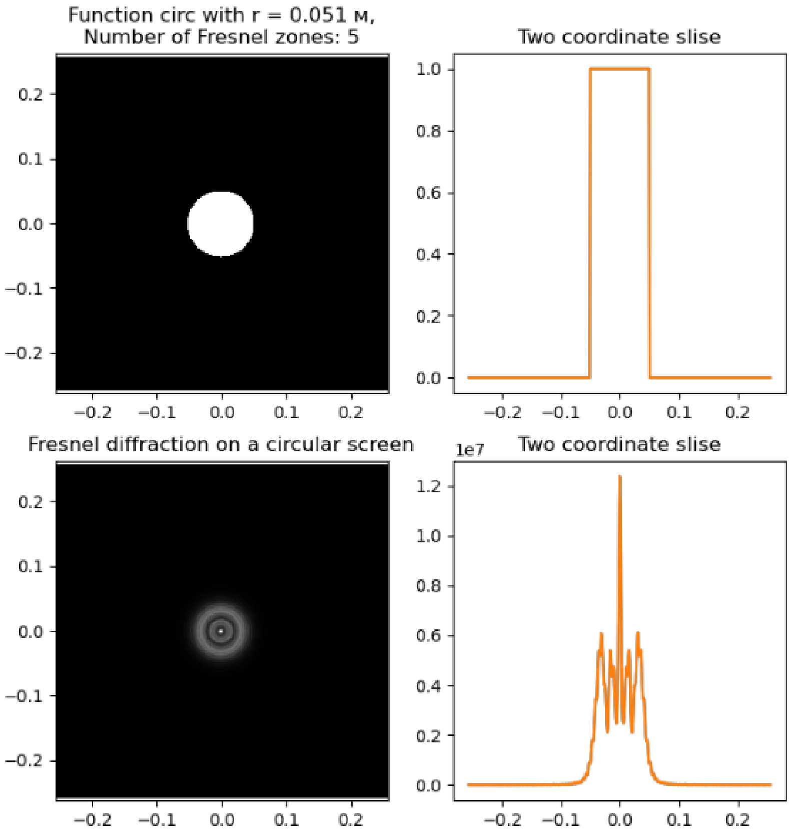

Рисунок 3 – Полутоновые изображения входной функции и распределение интенсивности на экране и их графики в сечениях по осям x и y

Задание 4:

Исследовать влияние ширины щели на распределение интенсивности в опыте Юнга. Вычислить аналитически и численно.

Аналитическое решение:

$$f(x,y) = rect\left(\frac{x-1}{b}\right) + rect\left(\frac{x+1}{b}\right), b - щель(мм)$$

$$F[f(x,y)] = u \cdot \delta(v_y) \cdot \sin c(\pi v_x) \cdot (\exp(2\pi i v_x) + \exp(-2\pi i v_x)) = 2b \cdot \delta(v_y) \cdot \sin c(\pi v_x) \cdot \cos(2\pi v_x)$$

$$I = \left|F[f(x,y)]\right|^2 = 4b^2 \cdot \left(\delta(v_y) \cdot \sin c(\pi v_x) \cdot \cos(2\pi v_x)\right)^2$$

Рассмотрим дифракционные картины для 5 случаев с шириной щели b=0.1-0.5mm:

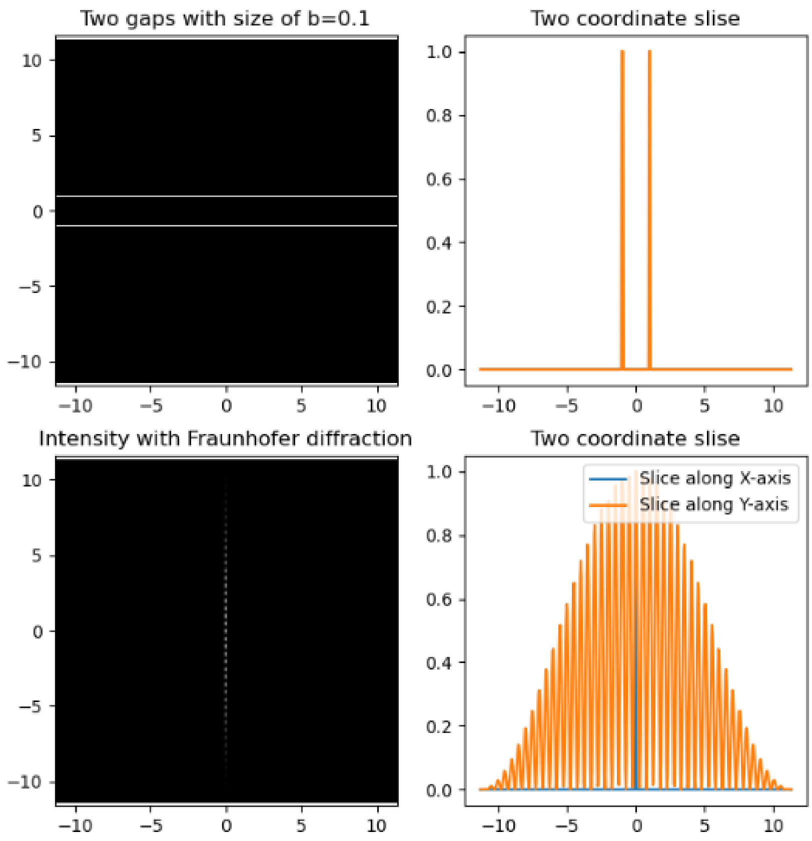

Рисунок 4 – Полутоновые изображения входной функции и распределение интенсивности на экране и их графики в сечениях по осям x и y

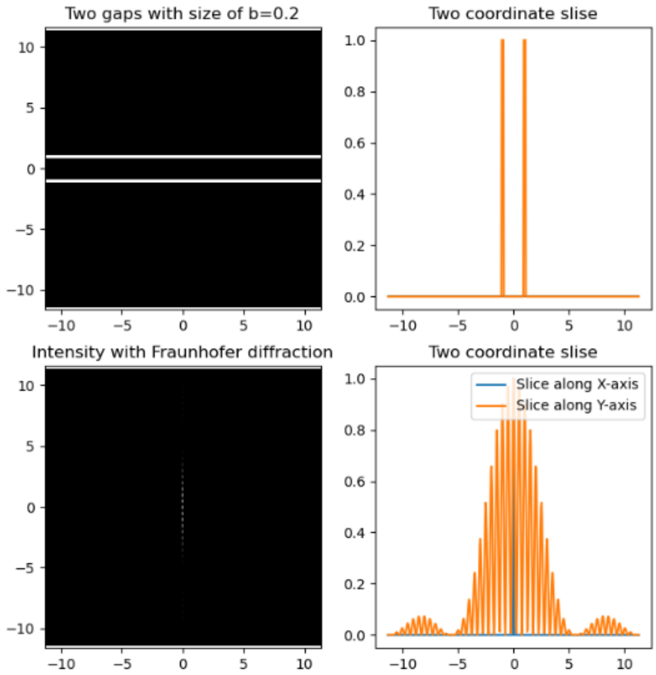

Рисунок 5 – Полутоновые изображения входной функции и распределение интенсивности на экране и их графики в сечениях по осям x и y

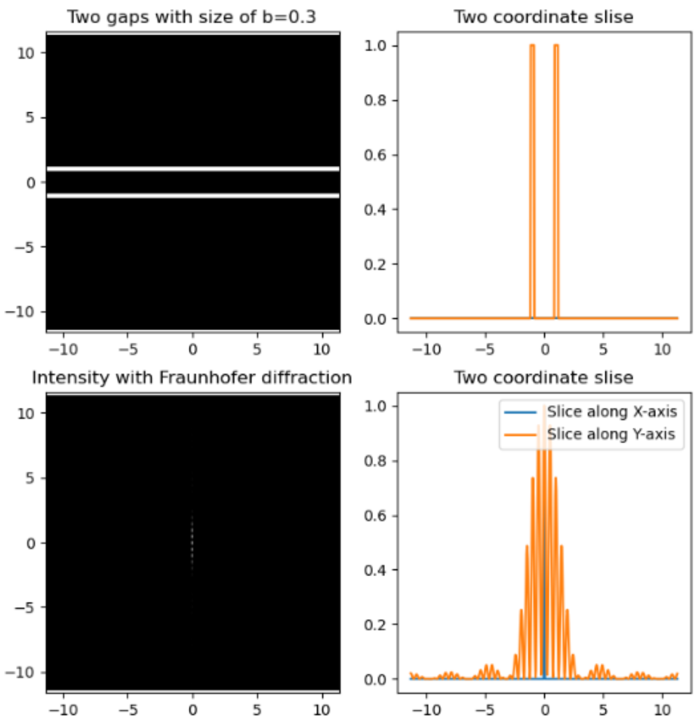

Рисунок 6 – Полутоновые изображения входной функции и распределение интенсивности на экране и их графики в сечениях по осям x и y

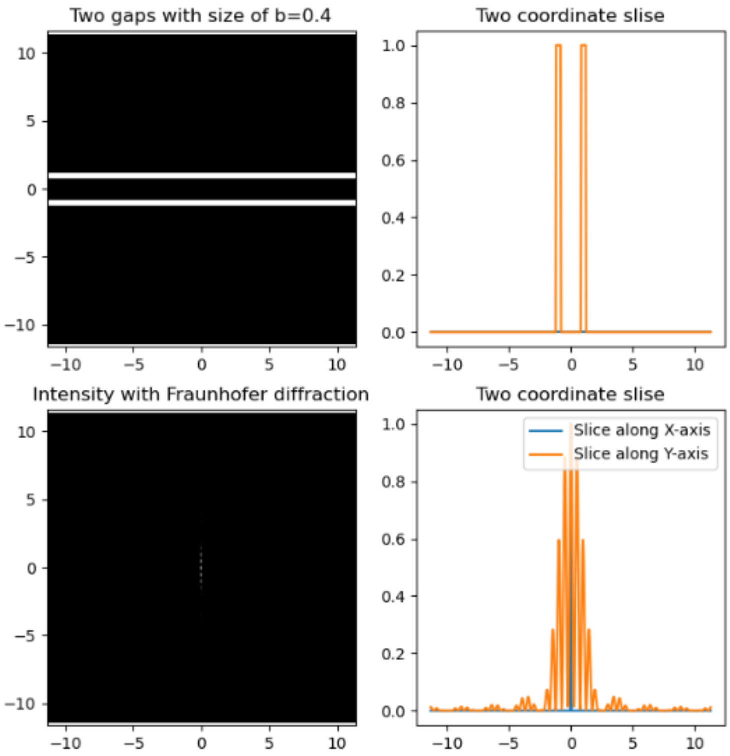

Рисунок 7– Полутоновые изображения входной функции и распределение интенсивности на экране и их графики в сечениях по осям x и y

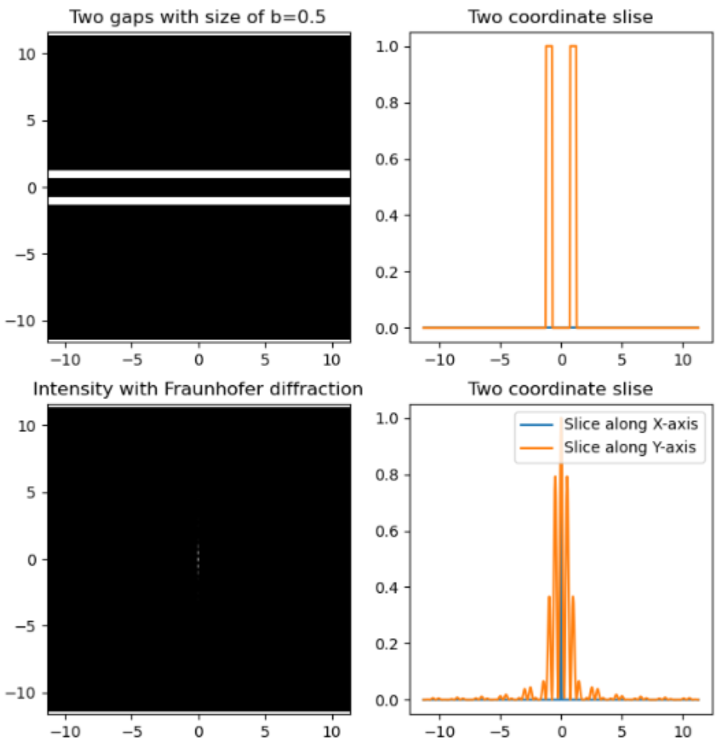

Рисунок 8 – Полутоновые изображения входной функции и распределение интенсивности на экране и их графики в сечениях по осям x и y

Вывод:

В ходе работы изучено моделирование дифракционных картин с использованием преобразования Фурье.

Текст программы:

```
import numpy as np
import matplotlib.pyplot as plt


def func_delta2(N, step_1mm):
    delta = np.zeros((N, N))
    delta[N//2 + 1 + step_1mm, :] = 1
    return delta


def func_circ(x, y, N, r):
    f_circ = np.zeros((N, N))
    for i in range(N):
        for j in range(N):
            if x[i, j]**2 + y[i, j]**2 <= r**2:
                f_circ[i, j] = 1
    return f_circ


def func_h(x, y, N, lambda_val, z, n, k):
    f_h = np.zeros((N, N), dtype=np.complex128)
    for i in range(N):
        for j in range(N):
            f_h[j, i] = (1/(1j*lambda_val*z)) * np.exp(1j*k*n*z)
* np.exp(1j*k*n*z) * np.exp(1j*k*n/(2*z)*(x[i, j]**2 + y[i,
j]**2))
    return f_h


def func_rect(x, N):
    f_rect = np.zeros((N, N))
    for i in range(N):
        for j in range(N):
            if -0.5 < x[i, j] < 0.5:
                f_rect[j, i] = 1
    return f_rect


def plot2(f_func, intensity, N, x, y, x_max, title1, title2):

    plt.figure(figsize=(12, 8))

    plt.subplot(231)
    plt.pcolor(x, y, f_func, cmap='gray')
```

```
    plt.axis('equal')
    plt.axis([-x_max, x_max, -x_max, x_max])
    plt.title(title1)

    plt.subplot(232)
    plt.plot(x[N//2, :], f_func[N//2, :], y[:, N//2], f_func[:,
N//2])
    plt.title("Two coordinate slise")

    plt.subplot(234)
    plt.pcolor(x, y, intensity, cmap='gray')
    plt.axis('equal')
    plt.axis([-x_max, x_max, -x_max, x_max])
    plt.title(title2)

    plt.subplot(235)
    plt.plot(x[N // 2, :], intensity[N // 2,
:]/np.max(intensity), label='Slice along X-axis')
    plt.plot(y[:, N // 2], intensity[:, N //
2]/np.max(intensity), label='Slice along Y-axis')
    plt.legend()
    plt.title("Two coordinate slise")



    plt.show()


def func124(f_func, N, x, y, x_max, title1, title2):
    f_func_shift = np.fft.fftshift(f_func)
    f_func_fft = np.fft.fft2(f_func_shift)
    f_func_fft_shift = np.fft.fftshift(f_func_fft)

    f_func_fft_shift = f_func_fft_shift / N
    intensity = np.abs(f_func_fft_shift)**2
    plot2(f_func, intensity, N, x, y, x_max, title1, title2)


def func3(f_func1, f_func2, N, x, y, x_max, title1, title2):
    f_func1_shift = np.fft.fftshift(f_func1)
    f_func1_fft = np.fft.fft2(f_func1_shift)
    f_func2_shift = np.fft.fftshift(f_func2)
    f_func2_fft = np.fft.fft2(f_func2_shift)

    f_funcs_fft = f_func1_fft * f_func2_fft
    f_funcs_fft_shift_ifft = np.fft.ifft2(f_funcs_fft)
    f_funcs_fft_shift_ifft_shift =
np.fft.fftshift(f_funcs_fft_shift_ifft)
    f_funcs_fft_shift_ifft_shift = f_funcs_fft_shift_ifft_shift
/ N
    intensity = np.abs(f_funcs_fft_shift_ifft_shift) ** 2
    plot2(f_func1, intensity, N, x, y, x_max, title1, title2)
```

```
def func_rect2(x, y, N):
    f_rect = np.zeros((N, N))
    for i in range(N):
        for j in range(N):
            if -0.5 < x[i, j] < 0.5 and -0.5 < y[i, j] < 0.5:
                f_rect[j, i] = 1
            else:
                f_rect[j, i] = 0
    return f_rect


# Basic data
N = 512
step = np.sqrt(1/N)
x_max = step * N/2

# filling x, y
x, y = np.meshgrid(np.arange(-x_max, x_max, step), np.arange(-
x_max, x_max, step))

# First part
f_func = func_rect2(x, 2 * y, N)
title1 = "Gap by Rect function"
title2 = "Intensity with Fraunhofer diffraction"
func124(f_func, N, x, y, x_max, title1, title2)  # for direct
Fourier

# Second part
step_1mm = int(np.ceil(1 / step))   # number of counts for +/- 1
mm
f_func = func_delta2(N, step_1mm) + func_delta2(N, -step_1mm)
title1 = "Two narrow gaps"
title2 = "Intensity with Fraunhofer diffraction"
func124(f_func, N, x, y, x_max, title1, title2)

# Third part
r = 0.051
z = 1000
n = 1
lambda_val = 0.5e-6
x_max = 0.256

k = 2 * np.pi / lambda_val
step = x_max * 2 / N
x, y = np.meshgrid(np.arange(-x_max, x_max, step), np.arange(-
x_max, x_max, step))
f_func = func_circ(x, y, N, r)
f_h = func_h(x, y, N, lambda_val, z, n, k)
num_zones_Fresnel = np.power(r, 2) / (lambda_val * z)  # number
of Fresnel zones
title1 = f"Function circ with r = 0.051 м,\nNumber of Fresnel
zones: {num_zones_Fresnel:.0f}"
```

```
title2 = "Fresnel diffraction on a circular screen"
func3(f_func, f_h, N, x, y, x_max, title1, title2)

# Fourth part
step = np.sqrt(1/N)
x_max = step * N/2
x, y = np.meshgrid(np.arange(-x_max, x_max, step), np.arange(-
x_max, x_max, step))
for i in range(1, 6):
    b = i * 1e-01
    f_func = func_rect((x + 1) / b, N) + func_rect((x - 1) / b,
N)
    title1 = f"Two gaps with size of b={b:.1f}"
    title2 = "Intensity with Fraunhofer diffraction"
    func124(f_func, N, x, y, x_max, title1, title2)
```