

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ**

Санкт-Петербургский национальный исследовательский университет
информационных технологий, механики и оптики

Мегафакультет трансляционных информационных технологий

Факультет информационных технологий и программирования

Лаб моделирование

Тема: Оптика, квантовая и атомная физика

Выполнила студент группы №М32111

Чу Тхи Фыонг Тхао

Проверил

Крылов В. А.

САНКТ-ПЕТЕРБУРГ

2021

Однощелевой эксперимент

Задаем функцию для вычисления интенсивности в diffraction.py

```
import numpy as np
# single slit diffraction
def single_slit_diffraction_intensity (slit_width, wavelength, screen_distance
, X):

    return ((np.sin((np.pi*slit_width*X)/(wavelength*screen_distance)))/((np.pi*
slit_width*X)/(wavelength*screen_distance)))**2
```

В файле single_slit_diffraction.py сначала импортируем используемые модули:

```
import matplotlib.pyplot as plt
import numpy as np
from matplotlib.widgets import Slider
from diffraction import single_slit_diffraction_intensity
```

Задаем пределы и шаг изменения координаты x:

```
X = np.arange(-0.005,0.005,0.00001)
```

Задаем длину волны, ширину щели и расстояние до экрана

```
wavelength = 400*(10**-9)
slit_width = 100*(10**-6)
screen_distance = 50*(10**-2)
```

Определяем Y как амплитуду:

```
Y = single_slit_diffraction_intensity(slit_width, wavelength, screen_distance, X)
```

Вызываем график и определяем оси:

```
plot, = plt.plot(X,Y)
plt.xlabel("Distance from center")
plt.ylabel("Intensity")
axis=(plt.axes([0.75, 0.75, 0.14, 0.05]))
axis2 = (plt.axes([0.75,0.65, 0.14, 0.05]))
axis3 = (plt.axes([0.75,0.55, 0.14, 0.05]))
```

Добавляем слайдер, чтобы можно было менять значения прямо на графике:

```
wavelength_slider = Slider(axis, 'Wavelength(nm)', 100, 1000, valinit=wavelength*10**9)
```

```
slit_width_slider = Slider(axis2, "Slit Width(micrometers)", 10, 1000, valinit=slit_width*10**6)
screen_distance_slider = Slider(axis3, "Screen Distance(cm)", 10, 100, valinit= screen_distance*10**2)
```

Добавляем обновление для действий слайдера и выводим график интенсивности:

```
def update(val) :
    wavelength = wavelength_slider.val*(10**-9)
    slit_width = slit_width_slider.val*(10**-6)
    screen_distance = screen_distance_slider.val*(10**-2)
    Y = single_slit_diffraction_intensity(slit_width, wavelength, screen_distance, X)
    plot.set_ydata(Y)

wavelength_slider.on_changed(update)
slit_width_slider.on_changed(update)
screen_distance_slider.on_changed(update)

plt.show()
```

Результат виртуального эксперимента:

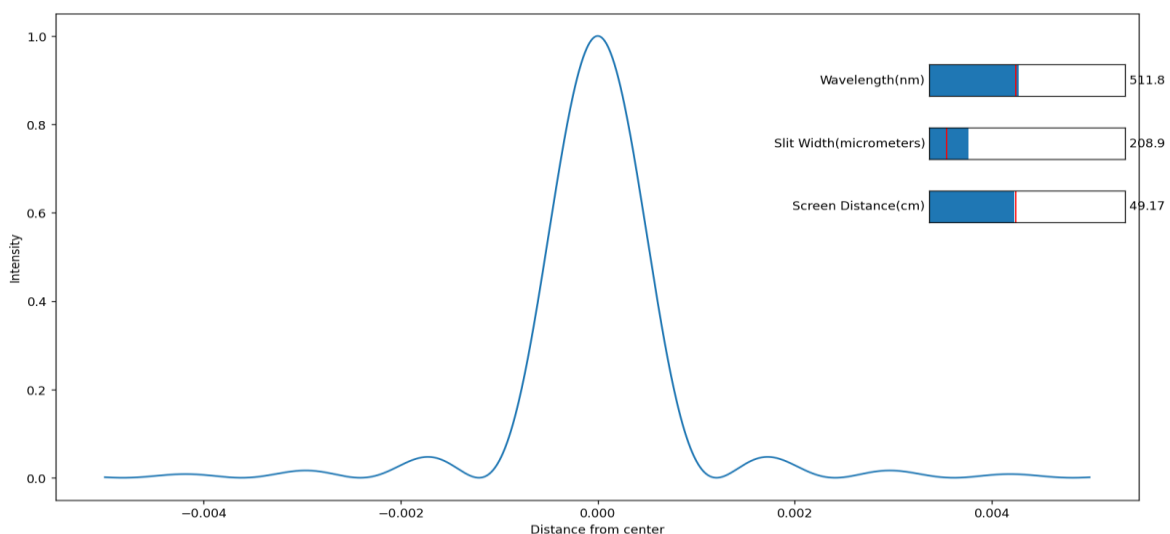


Рис. 1. Однощелевой эксперимент

Двухщелевой эксперимент

Задаем функцию для вычисления интенсивности в diffraction.py

```
def double_slit_diffraction_intensity (slit_width, wavelength, screen_distance
, distance_between_slits, X) :

    return (((np.sin((np.pi*slit_width*X)/(wavelength*screen_distance)))/((np.pi
*slit_width*X)/(wavelength*screen_distance)))**2)*((np.cos((np.pi*distance_bet
ween_slits*X)/(wavelength*screen_distance)))**2)
```

В файле double_slit_diffraction.py сначала импортируем используемые модули:

```
import matplotlib.pyplot as plt
import numpy as np
from matplotlib.widgets import Slider
from diffraction import double_slit_diffraction_intensity
```

Задаем пределы и шаг изменения координаты x:

```
X = np.arange(-0.005,0.005,0.00001)
```

Задаем длину волны, ширину щели, расстояние до экрана и расстояние между щелями:

```
slit_width = 100*(10**-6)
wavelength = 500*(10**-9)
screen_distance = 50*(10**-2)
distance_between_slits= 1*10**-3
```

Определяем Y как амплитуду:

```
Y = double_slit_diffraction_intensity(slit_width, wavelength, scr
een_distance, distance_between_slits, X)
```

Вызываем график и определяем оси:

```
plot, = plt.plot(X,Y)
```

```
plt.xlabel("Distance from center")
plt.ylabel("Intensity")

axis=(plt.axes([0.75, 0.75, 0.14, 0.05]))
axis2 = (plt.axes([0.75,0.65, 0.14, 0.05]))
axis3 = (plt.axes([0.75,0.55, 0.14, 0.05]))
axis4 = (plt.axes([0.75,0.45, 0.14, 0.05]))
```

Добавляем слайдер, чтобы можно было менять значения прямо на графике:

```
wavelength_slider = Slider(axis,'Wavelength(nm)',100, 1000, valinit=wavelength*10**9)
slit_width_slider = Slider(axis2, "Slit Width(micrometers)", 10, 1000, valinit=slit_width*10**6)
screen_distance_slider = Slider(axis3, "Screen Distance(cm)", 10, 100, valinit= screen_distance*10**2)
distance_between_slits_slider = Slider(axis4, "Distance b/w slits (mm)", 0.1, 10, valinit=distance_between_slits*10**3)
```

Добавляем обновление для действий слайдера и выводим график ИНТЕНСИВНОСТИ:

```
def update(val) :
    wavelength = wavelength_slider.val*(10**-9)
    slit_width = slit_width_slider.val*(10**-6)
    screen_distance = screen_distance_slider.val*(10**-2)
    distance_between_slits = distance_between_slits_slider.val*(10**
*-3)
    Y = double_slit_diffraction_intensity(slit_width, wavelength, s
creen_distance, distance_between_slits, X)
    plot.set_ydata(Y)

wavelength_slider.on_changed(update)
slit_width_slider.on_changed(update)
screen_distance_slider.on_changed(update)
distance_between_slits_slider.on_changed(update)

plt.show()
```

Результат виртуального эксперимента:

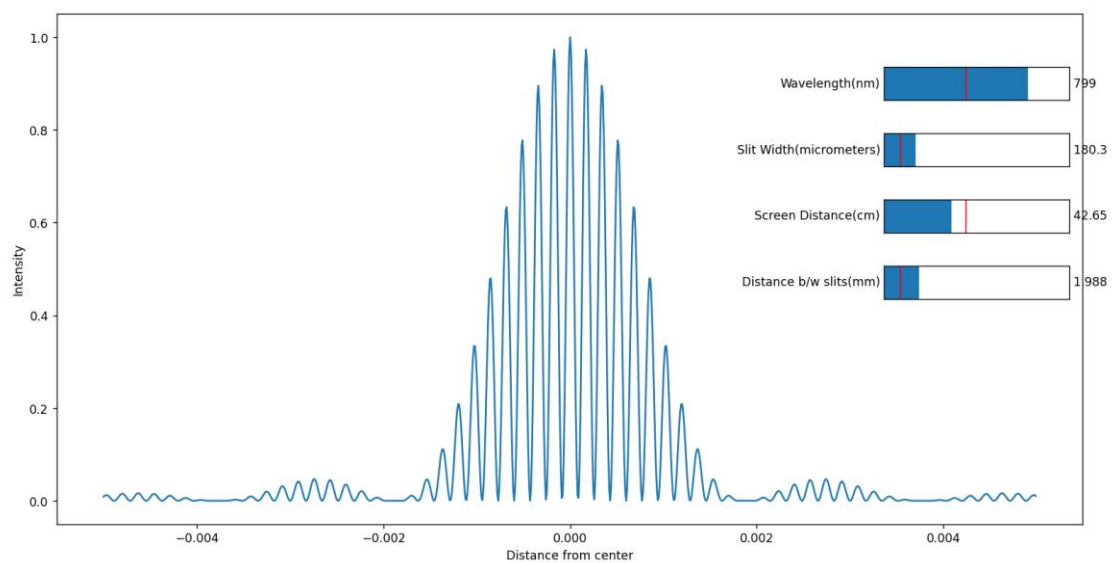


Рис. 2. Двухщелевой эксперимент