

УДК 535.542.1; 535.015

https://doi.org/10.33619/2414-2948/104/02

МОДЕЛИРОВАНИЕ ОПРЕДЕЛЕНИЯ ДЛИНЫ СВЕТОВОЙ ВОЛНЫ С ПОМОЩЬЮ ДИФРАКЦИОННОЙ РЕШЕТКИ НА PYTHON

©**Матисаков Ж. К.**, ORCID: 0009-0000-7643-5324, SPIN-код: 1645-4155, канд. техн. наук,
Ошский технологический университет им. М. Адышева,
г. Ош, Кыргызстан, mzhkka@gmail.com

©**Мамбетов Ж. И.**, ORCID: 0000-0003-4455-5887, SPIN-код: 7039-2541, канд. физ.-мат.
наук, Ошский технологический университет им. М. Адышева, г. Ош, Кыргызстан

©**Матисаков Ж. К.**, SPIN-код: 9544-8392, Ошский технологический университет
им. М. Адышева, г. Ош, Кыргызстан

MODELING THE DETERMINATION OF WAVELENGTH USING A DIFFRACTION GRATING IN PYTHON

©**Matisakov Zh.**, ORCID: 0009-0000-7643-5324, SPIN-code: 1645-4155, Ph.D.,
Osh Technological University after M. Adyshev, Osh, Kyrgyzstan, mzhkka@gmail.com

©**Mambetov Zh.**, ORCID: 0000-0003-4455-5887, SPIN-code: 7039-2541, Ph.D.,
Osh Technological University after M. Adyshev, Osh, Kyrgyzstan

©**Matisakov Zh.**, SPIN-code: 9544-8392, Osh Technological University after M. Adyshev,
Osh, Kyrgyzstan, mzhkka@gmail.com

Аннотация. Для моделирования лабораторной работы по определению длины световой волны с использованием дифракционной решетки на Python предлагается использовать библиотеки `numpy`, `matplotlib` и `ipywidgets`. Создается симуляция для моделирования положения максимумов на экране. Разработанный код может быть запущен в среде, поддерживающей виджеты `ipywidgets`, такой как Jupyter Notebook, для получения интерактивной симуляции определения длины световой волны с помощью дифракционной решетки. В коде определены следующие шаги: импорт необходимых библиотек: `numpy` для численных расчетов, `matplotlib` для построения графиков, `ipywidgets` для создания интерактивных виджетов; определение констант: d — расстояние между щелями дифракционной решетки, L — расстояние от решетки до экрана; создание функции `calculate_maxima` для расчета позиций максимумов; создание функции `plot_diffraction_pattern` для построения дифракционной картины; создание интерактивных виджетов: `wavelength_slider` для изменения длины волны света, `order_max_slider` для изменения максимального порядка интерференционных максимумов; запуск интерактивной симуляции, которая позволяет изменять параметры и наблюдать результаты в реальном времени.

Abstract. To model a laboratory experiment for determining the wavelength of light using a diffraction grating in Python, it is proposed to use the libraries `numpy`, `matplotlib`, and `ipywidgets`. A simulation is created to model the positions of maxima on the screen. The developed code can be run in an environment that supports `ipywidgets`, such as Jupyter Notebook, to obtain an interactive simulation for determining the wavelength of light using a diffraction grating. The code includes the following steps: 1. Import the necessary libraries: `numpy` for numerical calculations, `matplotlib` for plotting graphs, `ipywidgets` for creating interactive widgets. 2. Define constants: d - the distance between the slits of the diffraction grating, L - the distance from the grating to the screen. 3. Create the function `calculate_maxima` to calculate the positions of the maxima. 4. Create the function `plot_diffraction_pattern` to plot the diffraction pattern. 5. Create interactive widgets:

wavelength_slider to change the wavelength of light, order_max_slider to change the maximum order of the interference maxima. 6. Run the interactive simulation, which allows parameters to be adjusted and results to be observed in real-time.

Ключевые слова: дифракционная решетка, интерференция, максимумы, длина волны, моделирование, Python.

Keywords: diffraction grating, interference, maxima, wavelength, simulation, Python.

Дифракционная решетка представляет собой оптическое устройство, состоящее из множества узких параллельных щелей или рифлей, расположенных на некотором расстоянии друг от друга. Когда параллельный пучок света проходит через дифракционную решётку, каждая щель или рифль действует как источник вторичных волн, и эти вторичные волны интерферируют между собой [1].

Для анализа дифракционной решетки обычно используется условие интерференции для максимумов и минимумов интенсивности. Для максимума интерференции (яркий максимум) для дифракционной решетки, имеющей N широких параллельных щелей с шириной щели d и расстоянием между соседними щелями D , условие интерференции записывается следующим образом [1]:

$$d \sin(\theta) = m \lambda \quad (1)$$

где: d — ширина каждой щели, θ — угол, под которым наблюдается максимум интерференции, m — порядок интерференции (целое число), λ — длина волны света.

Если $m=0$, это соответствует центральному максимуму, а если $m=\pm 1, \pm 2, \pm 3, \dots$, это соответствует боковым максимумам.

Таким образом, для определения длины волны света с помощью дифракционной решётки необходимо измерить угол θ под которым наблюдается интерференционный максимум для заданного порядка интерференции m и известные параметры решетки d и D . Подставив измеренные значения в уравнение, можно рассчитать длину волны света λ .

Материал и методы исследования

Для создания симуляции лабораторной работы по определению длины световой волны с помощью дифракционной решетки на Python, можно использовать библиотеки `numpy`, `matplotlib` и `ipywidgets` для визуализации и интерактивного управления параметрами [2-4].

Будем использовать уравнение (1) для моделирования положения максимумов на экране и реализуем код.

Программный код состоит из следующих блоков:

1. Импорт библиотек:

- `numpy` для численных расчетов.
- `matplotlib` для построения графиков.
- `ipywidgets` для создания интерактивных виджетов.

2. Константы:

- d — расстояние между щелями дифракционной решетки.
- L — расстояние от решетки до экрана.

3. Функция `calculate_maxima`:

- Вычисляет углы θ для заданных порядков максимумов.
- Преобразует эти углы в позиции на экране.

4. Функция `plot_diffraction_pattern`:

- Использует `calculate_maxima` для получения позиций максимумов.
- Строит вертикальные линии на графике для визуализации дифракционной картины.

5. Интерактивные виджеты:

- `wavelength_slider` позволяет изменять длину волны света.
- `order_max_slider` позволяет изменять максимальный порядок интерференционных максимумов.

6. Интерактивная симуляция:

- `interact` из `ipywidgets` связывает функции и виджеты, что позволяет интерактивно изменять параметры и видеть результаты в реальном времени.

Результаты и обсуждение

Запустим код в Jupyter Notebook или другой среде, поддерживающей виджеты `ipywidgets`, чтобы получить интерактивную симуляцию определения длины световой волны с помощью дифракционной решетки. Программа позволяет изменять длину волны и наблюдать, как это влияет на положение дифракционных максимумов.

На Рисунках 1-3 представлены результаты работы программы. Используя интерактивные виджеты `wavelength_slider` и `order_max_slider`, можно изменять соответственно длину волны света и максимальный порядок интерференционных максимумов, получая различные дифракционные картины.

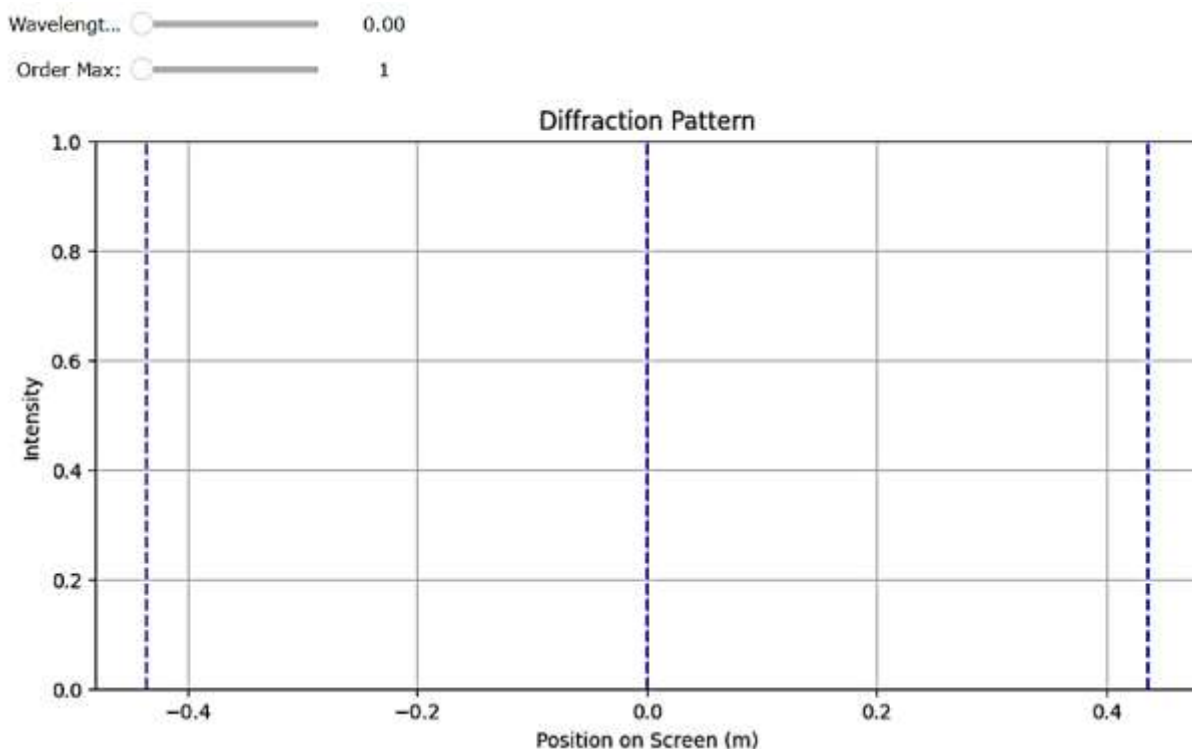


Рисунок 1. Дифракционная картина при значениях $\lambda = 550\text{нм}$, $m=1$

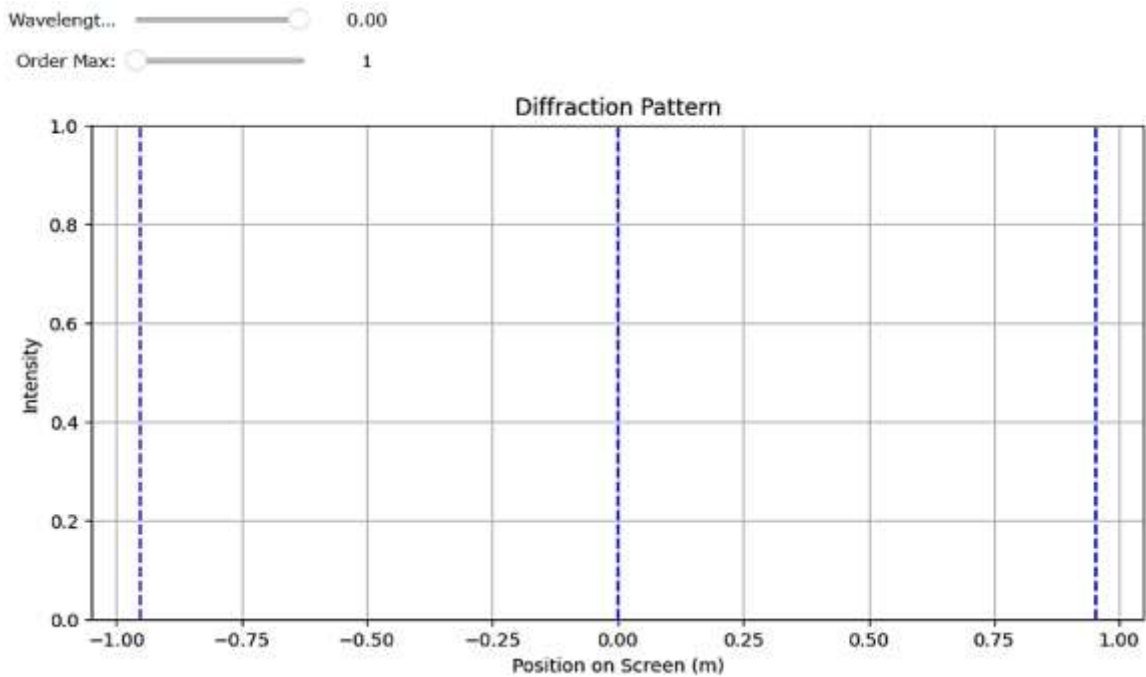


Рисунок 2. Дифракционная картина при значениях $\lambda = 700\text{nm}$, $m=1$

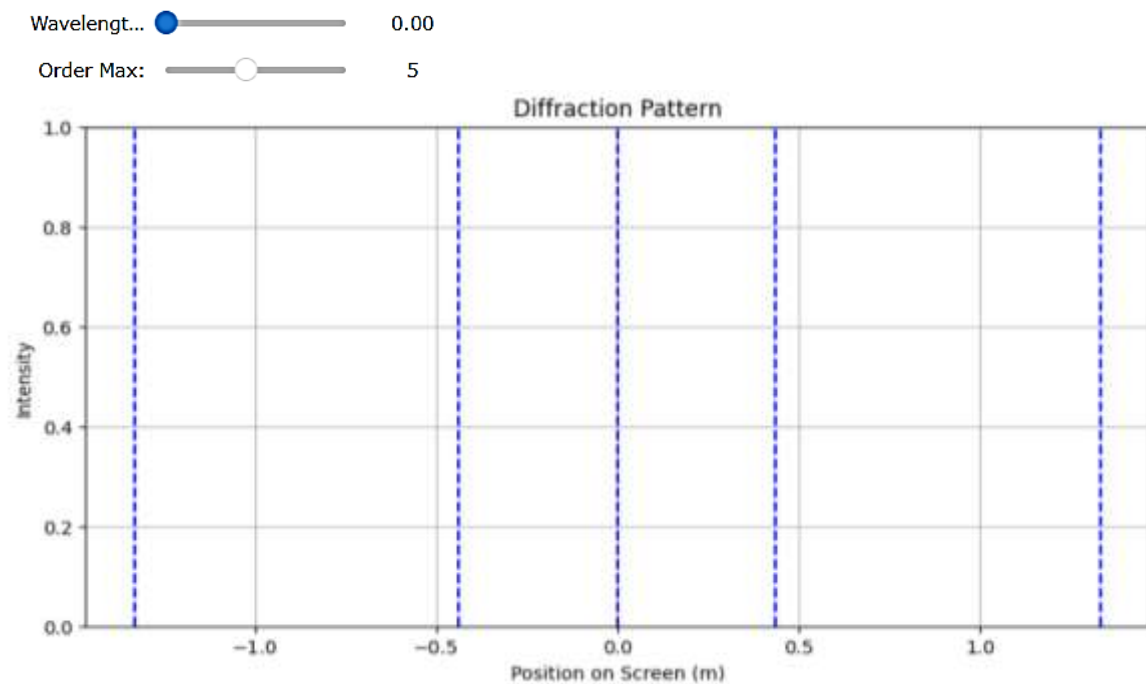


Рисунок 3. Дифракционная картина при значениях $\lambda = 400\text{nm}$, $m=5$

Заключение

Для моделирования лабораторной работы по определению длины световой волны с использованием дифракционной решетки на Python предлагается использовать библиотеки `numpy`, `matplotlib` и `ipywidgets`. Создается симуляция для отображения положений максимумов на экране. Разработанный код можно запускать в среде, поддерживающей виджеты `ipywidgets`, такой как Jupyter Notebook, чтобы получить интерактивную симуляцию определения длины световой волны с помощью дифракционной решетки. В коде выполняются следующие шаги:

1. Импорт необходимых библиотек: `numpy` для численных расчетов, `matplotlib` для построения графиков, `ipywidgets` для создания интерактивных виджетов.
2. Определение констант: d — расстояние между щелями дифракционной решетки, L — расстояние от решетки до экрана.
3. Создание функции `calculate_maxima` для расчета позиций максимумов.
4. Создание функции `plot_diffraction_pattern` для построения дифракционной картины.
5. Создание интерактивных виджетов: `wavelength_slider` для изменения длины волны света, `order_max_slider` для изменения максимального порядка интерференционных максимумов.
6. Запуск интерактивной симуляции, которая позволяет изменять параметры и наблюдать результаты в реальном времени.

Список литературы:

1. Трофимова Т. И. Курс физики. М.: Академия, 2006. 560 с.
2. Бабаев Д. Б., Матисаков Ж. К. Моделирование физических явлений и процессов в VPython // Бюллетень науки и практики. 2023. Т. 9. №7. С. 370-374. <https://doi.org/10.33619/2414-2948/92/51>
3. Бабаев Д. Б., Матисаков Ж. К. Создание виртуальных лабораторных работ по физике в VPython // Бюллетень науки и практики. 2023. Т. 9. №7. С. 375-378. <https://doi.org/10.33619/2414-2948/92/52>
4. Asanova S., Matisakov Zh. K., Atamkulova M. T. Application of training simulators in teaching programming // *Izvestiy Oshskogo technologicheskogo university*. 2018. №2. P. 104-107. EDN IDIARD.

References:

1. Trofimova, T. I. (2006). *Kurs fiziki*. Moscow. (in Russian).
2. Babaev, D., & Matisakov, Zh. (2023). Modeling Physical Phenomena and Processes in VPython. *Bulletin of Science and Practice*, 9(7), 370-374. (in Russian). <https://doi.org/10.33619/2414-2948/92/51>
3. Babaev, D., & Matisakov, Zh. (2023). Creating Virtual Physics Labs in VPython. *Bulletin of Science and Practice*, 9(7), 375-378. (in Russian). <https://doi.org/10.33619/2414-2948/92/52>
4. Asanova, S., Matisakov, Zh. K., & Atamkulova, M. T. (2018). Application of training simulators in teaching programming. *Izvestiy Oshskogo technologicheskogo university*, (2), 104-107.

*Работа поступила
в редакцию 10.06.2024 г.*

*Принята к публикации
17.06.2024 г.*

Ссылка для цитирования:

Матисаков Ж. К., Мамбетов Ж. И., Матисаков Ж. К. Моделирование определения длины световой волны с помощью дифракционной решетки на Python // Бюллетень науки и практики. 2024. Т. 10. №7. С. 21-25. <https://doi.org/10.33619/2414-2948/104/02>

Cite as (APA):

Matisakov, Zh., Mambetov, Zh., & Matisakov, Zh. (2024). Modeling the Determination of Wavelength Using a Diffraction Grating in Python. *Bulletin of Science and Practice*, 10(7), 21-25. (in Russian). <https://doi.org/10.33619/2414-2948/104/02>

