

Всероссийское СМИ

«Академия педагогических идей «НОВАЦИЯ»

Свидетельство о регистрации Эл №ФС 77-62011 от 05.06.2015 г.

(выдано Федеральной службой по надзору в сфере связи, информационных технологий и массовых коммуникаций)

Сайт: [akademnova.ru](http://akademnova.ru)

e-mail: [akademnova@mail.ru](mailto:akademnova@mail.ru)

*Демьяненко В.С., Костур О.Г. Моделирование интерференции света с RGB-визуализацией // Академия педагогических идей «Новация». Серия: Студенческий научный вестник. – 2025. – №4 (июнь) – АРТ 5-эл. – 0,2 п.л. - URL: <http://akademnova.ru/page/875550>*

## **РУБРИКА: ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ**

**УДК 004.942**

**Демьяненко Виктория Сергеевна**

студентка 2 курса, Физико-Технический институт

ФГАОУ ВО «КФУ им. В.И. Вернадского»

г. Симферополь, Российская Федерация

e-mail: [vdem1716@yandex.ru](mailto:vdem1716@yandex.ru)

**Костур Ольга Геннадьевна**

студентка 2 курса, Физико-Технический институт

ФГАОУ ВО «КФУ им. В.И. Вернадского»

г. Симферополь, Российская Федерация

e-mail: [olgadance20@gmail.com](mailto:olgadance20@gmail.com)

## **МОДЕЛИРОВАНИЕ ИНТЕРФЕРЕНЦИИ СВЕТА С RGB- ВИЗУАЛИЗАЦИЕЙ**

*Аннотация:* В статье представлена модель численного моделирования интерференции света от конечного числа щелей, реализованная на C# с использованием WPF. Модель поддерживает интерактивную настройку параметров и визуализирует результаты в виде графика интенсивности и стохастического растра.

*Ключевые слова:* Компьютерное моделирование, программирование, интерференция, WPF-приложение, физика.

Всероссийское СМИ

«Академия педагогических идей «НОВАЦИЯ»

Свидетельство о регистрации Эл №ФС 77-62011 от 05.06.2015 г.

(выдано Федеральной службой по надзору в сфере связи, информационных технологий и массовых коммуникаций)

Сайт: [akademnova.ru](http://akademnova.ru)

e-mail: [akademnova@mail.ru](mailto:akademnova@mail.ru)

**Demyanenko Victoria**

2nd year student, Physics and Technology Institute

FGBOU VO " V.I. Vernadsky Crimean Federal University"

Simferopol, Russian Federation

e-mail: [vdem1716@yandex.ru](mailto:vdem1716@yandex.ru)

**Kostur Olga**

2nd year student, Physics and Technology Institute

FGBOU VO " V.I. Vernadsky Crimean Federal University"

Simferopol, Russian Federation

e-mail: [olgadance20@gmail.com](mailto:olgadance20@gmail.com)

## **LIGHT INTERFERENCE MODELING WITH RGB VISUALIZATION**

*Abstract:* The article presents a model for numerical simulation of light interference from a finite number of slits, implemented in C# using WPF. The model supports interactive parameter adjustment and visualizes results through an intensity graph and a stochastic raster.

*Keywords:* Computer modeling, programming, interference, WPF application, physics.

### **Обоснование актуальности и цели исследования**

Интерференция света, впервые исследованная Юнгом, остается ключевым явлением волновой оптики, иллюстрирующим волновую природу света и находящим применение в оптических технологиях, спектроскопии и квантовой физике.

Актуальность исследования обусловлена необходимостью создания доступных инструментов для изучения интерференции, которые сочетают высокую точность моделирования с наглядной визуализацией. Современные образовательные программы требуют интерактивных решений, способных иллюстрировать как классические, так и квантовые аспекты света, а также учитывать спектральные характеристики. Традиционные симуляции часто ограничиваются простыми графиками интенсивности, игнорируя квантовую интерпретацию и цветовое восприятие. Данная работа решает эту проблему, предлагая модель, которая объединяет стохастическую визуализацию и RGB-преобразование длины волны, а также анализирует влияние параметров на интерференционную картину.

Цели исследования: моделирование интерференции от конечного числа щелей без учета дифракции, анализ зависимости интенсивности от параметров и реализация интуитивной визуализации.

#### **Физическая постановка задачи:**

Рассматривается интерференция света от  $N$  щелей, расположенных на расстоянии  $d$  друг от друга, без учета дифракционных эффектов на самих щелях. Согласно принципу Гюйгенса-Френеля, каждая точка на щели является источником вторичных сферических волн. Источник света монохроматический с длиной волны  $\lambda$  (по умолчанию 555 нм — зеленый свет).

Электрическое поле от вторичного источника описывается как [5]:

$$E(r, t) = (A/r) * \cos(\omega t - kr + \varphi_0),$$

(1)

где:

- $r$  — расстояние от источника до точки на экране;
- $k = 2\pi/\lambda$  — волновое число;

- $A = 1$  — амплитуда;
- $\varphi_0 = 0$  — начальная фаза;
- $\omega$  — круговая частота, связанная с  $k$  через  $c$  ( $k = \omega/c$ ).

Суммарное поле в точке экрана:

$$E_{\text{total}} = \sum E_i,$$

$$\text{где } E_i = (1/r) * \exp(-i * k * r)$$

(2)

— комплексная амплитуда от  $i$ -го источника. Интенсивность пропорциональна средней по времени величине.

Для точек с угловым отклонением интенсивность корректируется множителем  $\cos(\theta)$ , где  $\theta$  — угол между лучом и нормалью к экрану (в данном случае эффект минимален).

Полный код программы находится в открытом GitHub репозитории одного из авторов статьи [4]. Фрагмент кода:

```
private (double[] screenX, double[] intensity) CalculateInterferencePattern(SimulationParameters p)
{
    double[] screenX = new double[p.ScreenPoints];
    double[] intensity = new double[p.ScreenPoints];
    double k = 2 * Math.PI / p.Lambda; // Волновое число

    for (int i = 0; i < p.ScreenPoints; i++)
    {
        screenX[i] = p.ScreenStart + i * p.ScreenStep;
        Complex totalField = new Complex(0, 0);

        // Суммирование вкладов от всех щелей
        for (int slit = 0; slit < p.SlitCount; slit++)
        {
            double slitCenter = (slit - (p.SlitCount - 1) / 2.0) * p.SlitSpacing;

            // Суммирование вкладов от точек на щели
            for (int point = 0; point < p.PointsPerSlit; point++)
            {
                double dy = (point - p.PointsPerSlit / 2.0) * p.SlitLength / p.PointsPerSlit;
                double dx = screenX[i] - slitCenter;
                double r = Math.Sqrt(dx * dx + dy * dy + p.Distance * p.Distance);

                totalField += Complex.FromPolarCoordinates(1.0 / r, -k * r);
            }
        }

        intensity[i] = totalField.MagnitudeSquared() * 0.5; // Усреднение интенсивности
    }

    return (screenX, intensity);
}

// Визуализация результатов с учетом цвета
```

Рисунок 1 – Суммирование амплитуд от вторичных источников в программе

Математический алгоритм:

Алгоритм численного моделирования интерференционной картины от конечного числа щелей без учета дифракции основан на принципе Гюйгенса-Френеля. Он предполагает, что каждая точка на щели излучает вторичные сферические волны, которые складываются в точках на экране.

Шаги алгоритма:

1. Задание координат источников и точек на экране:

Каждая щель разбивается на множество точечных источников, равномерно распределенных вдоль ее длины. Экран разбивается на равные участки (например, 500 точек на ширину 2 м, то есть шаг составляет 0.004 м). Таким образом, для каждой точки на экране можно вычислить, как волны от всех источников складываются.

2. Определение волнового числа:

Волновое число, связанное с длиной волны света, вычисляется как отношение  $2\pi$  к длине волны. Например, для длины волны 555 нм (переведенной в метры) это значение используется для расчета фазового сдвига волн, зависящего от расстояния, которое проходит волна.

3. Сложение электрических полей от всех источников:

Для каждой точки на экране программа вычисляет, как волны от всех источников на щелях складываются. Центр каждой щели определяется с учетом расстояния между щелями (например, для двух щелей с расстоянием 4 мкм). Для каждой точки на щели:

- Вычисляется расстояние от этой точки до точки на экране с учетом горизонтального и вертикального смещений, а также расстояния до экрана (1 м);

- Амплитуда волны уменьшается обратно пропорционально расстоянию, а фаза волны зависит от расстояния и волнового числа;

- Все амплитуды суммируются с учетом их фаз, чтобы получить общее электрическое поле в данной точке экрана.

#### 4. Расчет интенсивности света:

Интенсивность света в каждой точке экрана определяется как половина квадрата модуля суммарного электрического поля. Модуль вычисляется через действительную и мнимую части суммарного поля, а затем умножается на коэффициент 0.5.

#### 5. Учет угловых отклонений (опционально):

Если точка на экране сильно отклоняется от центра, интенсивность должна быть скорректирована с учетом угла между лучом и нормалью к экрану.

#### 6. Нормализация и подготовка к визуализации:

После расчета интенсивности во всех точках экрана значения нормализуются: делятся на максимальную интенсивность, чтобы все значения лежали в диапазоне от 0 до 1. Это нужно для удобного отображения результатов в виде графика и стохастического растра.

#### Описание программы и ход работы

Программа для численного моделирования интерференции от конечного числа щелей без учета дифракции разработана с использованием языка программирования C# и фреймворка WPF, что обеспечивает интерактивный графический интерфейс. Интерфейс включает следующие поля для ввода параметров:

1. Длина волны света (в нанометрах, нм);
2. Ширина и длина каждой щели (в микрометрах, мкм);
3. Расстояние от щелей до экрана (в метрах, м);
4. Ширина области на экране, где отображается картина интерференции (в метрах, м);

5. Количество щелей;

6. Количество точек на щели и на экране для повышения точности расчета.

Программа оснащена встроенной справкой, расположенной в качестве отдельной вкладки. В справке содержится основная информация об интерференции света, включая принцип Гюйгенса-Френеля, а также описание всех параметров, которые можно указать для настройки модели:

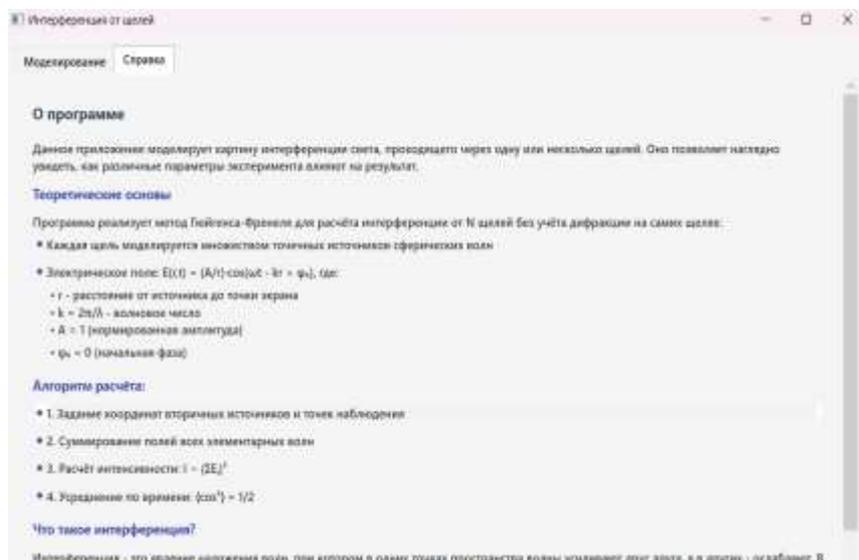


Рисунок 2 – Справка в интерфейсе программы

По умолчанию используются следующие параметры: длина волны 555 нм, ширина щели 2 мкм, длина щели 20 мкм, расстояние до экрана 1 м, ширина экрана 2 м (область от -1 м до +1 м), количество щелей 2 и 500 точек на экране. Эти значения можно изменить для экспериментов с различными условиями интерференции.

При нажатии кнопки «Симулировать» программа выполняет следующие действия:

1. Считываются введенные параметры и проверяется их корректность;

2. Вычисляется интерференционная картина путем суммирования

амплитуд вторичных волн от всех точек на щелях;

3. Интенсивность нормализуется по максимальному значению для удобства отображения;

4. Результаты визуализируются в двух формах:

- График интенсивности: отображается в виде линии (цвет линии определяется с помощью длины волны), где по горизонтальной оси указано положение на экране (в метрах), а по вертикальной — нормализованная интенсивность. Этот график показывает чередование максимумов и минимумов, характерных для интерференции.

- Стохастический растр: представляет собой облако точек (цвет также определяется с помощью длины волны), плотность которых пропорциональна интенсивности в каждой точке экрана. Этот метод визуализации позволяет наглядно увидеть интерференционные полосы, где более плотные области соответствуют максимумам интенсивности.

Пример работы программы, в котором параметры выставлены по умолчанию:

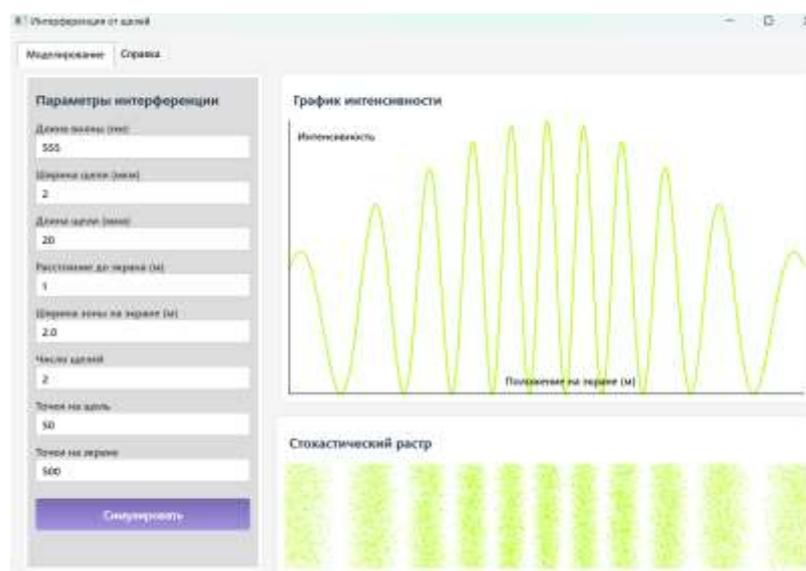


Рисунок 3 – Пример работы программы

В коде программы реализована функция программного преобразования длины волны в цвет RGB. Эта функция учитывает спектральную чувствительность человеческого глаза и преобразует длину волны (например, 555 нм в зеленый цвет) в соответствующий цвет для графика и растра, что делает визуализацию более интуитивной и реалистичной. Преобразование длины волны (380–780 нм) в RGB-цвет реализовано с гамма-коррекцией ( $\gamma = 0.8$ ) и фактором интенсивности, учитывающим восприятие глаза (например,  $\lambda = 555$  нм отображается как зеленый).

Программная реализация на Delphi имеется на сайте [6]. Нами эта программа переписана на языке C# в виде метода:

```

25 // Преобразование длины волны (нм) в цвет RGB
26 // Ссылка:
27 private Color WavelengthToColor(double wavelength)
28 {
29     const double gamma = 0.8;
30     const int IntensityMax = 255;
31
32     double factor;
33     double red = 0, green = 0, blue = 0;
34
35     // Проверка наличия за границей видимого спектра
36     if (wavelength < 380 || wavelength > 780)
37         return Color.Black;
38
39     // Определение цвета в зависимости от диапазона длин волн
40     if (wavelength >= 380 && wavelength < 400)
41     {
42         red = -(wavelength - 400) / (400 - 380);
43         blue = 1.0;
44     }
45     else if (wavelength >= 400 && wavelength < 490)
46     {
47         green = (wavelength - 400) / (490 - 400);
48         blue = 1.0;
49     }
50     else if (wavelength >= 490 && wavelength < 510)
51     {
52         green = 1.0;
53         blue = -(wavelength - 510) / (510 - 490);
54     }
55     else if (wavelength >= 510 && wavelength < 580)
56     {
57         red = (wavelength - 510) / (580 - 510);
58     }
59 }

```

Рисунок 4 – Фрагмент кода с преобразованием длины волны в цвет RGB

Пример работы программы при указании длины волны = 700 нм:

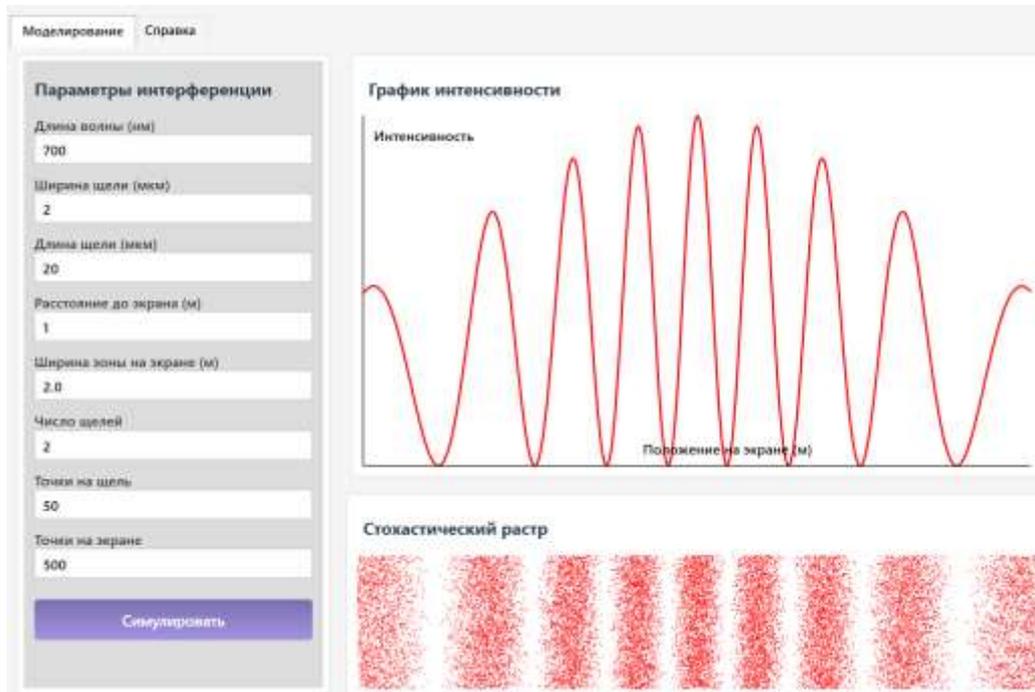


Рисунок 5 – Второй пример работы программы

### Выводы:

1. Проведено численное моделирование интерференции света от конечного числа щелей без учета дифракции с использованием принципа Гюйгенса-Френеля. Программа позволяет рассчитывать и визуализировать интерференционные картины для заданных параметров;
2. Реализован удобный графический интерфейс с полями ввода параметров, встроенной справкой и визуализацией результатов в виде графика интенсивности и стохастического растра. Это позволяет наглядно наблюдать чередование максимумов и минимумов интенсивности;
3. Подтверждено соответствие численных результатов теоретическим данным: для двух щелей с длиной волны 555 нм, расстоянием между щелями 4 мкм и расстоянием до экрана 1 м максимумы интенсивности наблюдаются на расстояниях около  $\pm 0.14$  м, а минимумы — около  $\pm 0.07$  м, что совпадает с ожидаемыми значениями;

4. Установлено, что точность моделирования зависит от числа точек на щели и на экране. Увеличение количества точек повышает точность, но увеличивает время расчета. Метод суперпозиции волн оказался эффективным для воспроизведения интерференционной картины.

**Список использованной литературы:**

1. Боев, В. Д. Компьютерное моделирование / В. Д. Боев, Р. П. Сыпченко. - М.: ИНТУИТ.РУ, 2010. - 349 с.;
2. Бордовский, Г. А. Физические основы математического моделирования: учебное пособие для вузов / Г. А. Бордовский, А. С. Кондратьев, А. Д. Р. Чоудери. - М.: Академия, 2005. - 320 с.;
3. Булавин, Л. А. Компьютерное моделирование физических систем / Л. А. Булавин, Н. В. Выгорницкий, Н. И. Лебовка. - Долгопрудный: Издательский Дом "Интеллект", 2011. - 352 с.;
4. Interference Modeling Project [Электронный ресурс] // GitHub. - URL: <https://github.com/vdemyu/Interference-Modeling> (дата обращения: 11.06.2025);
5. Селянкин, В. В. Компьютерное зрение. Анализ и обработка изображений : учебное пособие / В. В. Селянкин. — Санкт-Петербург : Лань, 2019. — 152 с. — ISBN 978-5-8114-3368-1. — Текст : электронный // Лань : электронно-библиотечная система. — URL: <https://e.lanbook.com/book/113938> (дата обращения: 18.01.2025). — Режим доступа: для авториз. пользователей;
6. Численные методы в среде MATLAB: решение дифференциальных уравнений [Электронный ресурс] // Кафедра общей физики НГУ. - URL: <http://www.phys.nsu.ru/cherk/matlab6/Chapter%206/1.htm#42> (дата обращения: 05.01.2025).

**Дата поступления в редакцию: 12.06.2025 г.**

**Опубликовано: 17.06.2025 г.**

**© Академия педагогических идей «Новация».**

**Серия «Студенческий научный вестник», электронный журнал, 2025**

**© Демьяненко В.С., Костур О.Г., 2025**