

МОСКОВСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

им. М.В. ЛОМОНОСОВА

ФИЗИЧЕСКИЙ ФАКУЛЬТЕТ

---

ДИПЛОМНАЯ РАБОТА

СТУДЕНТА 605 ГРУППЫ

ШАЛАГИНА АЛЕКСЕЯ ДМИТРИЕВИЧА

РАЗРАБОТКА И СОЗДАНИЕ КОМПЛЕКСА

ДЕМОНСТРАЦИОННЫХ ПРОГРАММ

ПО ВОЛНОВОЙ И КВАНТОВОЙ ОПТИКЕ

НА ОСНОВЕ МОДУЛЬНОЙ АРХИТЕКТУРЫ

Допущен к защите

« 23 » декабря 2011 г.

Заведующий кафедрой

общей физики

д.ф.-м.н, профессор

Научный руководитель:

ст. преподаватель кафедры

общей физики, к.п.н.

А.М. Салецкий

А.В. Селиверстов

МОСКВА, 2011

# Оглавление

<b>Введение</b>	<b>3</b>
<b>1 Методические основы создания демонстрационных модельных программ</b>	<b>5</b>
1.1 Дидактические и технические требования к модельным программам . . .	7
1.2 Психофизиологические и эргономические требования к модельным программам . . . . .	9
1.3 Операторский и демонстрационный интерфейсы . . . . .	12
1.4 Концепция интерфейса модельных демонстраций . . . . .	12
<b>2 Принципы построения и типовые элементы интерфейса демонстрационных модельных программ</b>	<b>14</b>
2.1 Библиотека QT . . . . .	15
2.2 Библиотека OpenGL . . . . .	17
2.3 Структуры, потоки данных и элементы интерфейса . . . . .	18
2.4 Отображение цвета монохроматического излучения . . . . .	20
<b>3 Созданные демонстрационные программы</b>	<b>21</b>
3.1 Программы по интерференции света . . . . .	21
3.1.1 Интерферометр Майкельсона . . . . .	21
3.1.2 Кольца Ньютона . . . . .	22
3.1.3 Схема Юнга . . . . .	24
3.1.4 Тонкая пластинка (полосы равного наклона) . . . . .	27
3.2 Программы по дифракции света . . . . .	28
3.2.1 Дифракция Френеля на круглом отверстии . . . . .	28
3.3 Основы теории дифракции Френеля . . . . .	32
3.3.1 Интеграл Гюйгенса–Френеля . . . . .	32
3.4 Дифракция Фраунгофера . . . . .	33

# Введение

Одним из основных общедидактических требований к лекциям по общей физике является требование наглядности изложения материала [9]. В качестве средств наглядности выступают различные демонстрационные материалы (плакаты, фотографии, видеозаписи и проч.), но основным традиционно остается демонстрационный эксперимент [8].

С 90-х годов XX века компьютерная техника все больше используется в процессе обучения. В связи с этим получили широкое распространение компьютерные модельные демонстрации, дополняющие или даже заменяющие натуральный эксперимент.

Модельные демонстрации свободны от многих недостатков натурального эксперимента [11]. Однако анализ существующих компьютерных программ, разработанных как в нашей стране, так и за рубежом, показал, что значительная их часть имеет недостатки и не в полной мере удовлетворяет требованиям, предъявляемым к средствам обеспечения учебного процесса. Это может быть следствие недостатков среды разработки, ее устаревания или неполного использования возможностей такой среды на момент создания программы. Большинство программных интерфейсов разработано без учета специфики демонстрационных программ [4]. Особенно это актуально для демонстраций по оптике, в которых наглядность представления информации и реалистичность ее воспроизведения играют важнейшую роль. В ближайшем будущем роль компьютерных программ в учебном процессе будет только расти. Поэтому необходимы программы нового поколения, в которых будут устранены такого рода недостатки.

В силу специфики компьютерных демонстраций при их написании на первый план выходят не проблемы реализации физической модели, а гораздо более трудоемкие задачи создания соответствующего интерфейса и его реализации. Поэтому для эффективной разработки таких программ необходима технология, объединяющая усилия многих программистов, позволяющая распределять написание программного кода и т.п. Это требует применения системного подхода к такой разработке, основывающегося в первую очередь на унификации архитектуры программ.

Целью настоящей работы являлась разработка архитектуры системы демонстрационных программ, а также создание новых демонстраций по волновой и квантовой оптике. В задачи работы входили:

- анализ структуры демонстрационных программ;
- выделение структур данных и алгоритмов их обработки;
- анализ потоков данных;
- создание шаблона программы;
- разработка комплекса демонстрационных программ, охватывающих различные области оптики, имеющих оптимальный для использования на лекциях по общей физике интерфейс и удовлетворяющих остальным требованиям к компьютерным демонстрациям.

# Глава 1

## Методические основы создания демонстрационных модельных программ

Во введении отмечалось, что модельные демонстрации имеют ряд преимуществ перед натурными. К основным из них можно отнести [11]:

- отсутствие необходимости длительной настройки;
- отсутствие необходимости специфического дорогостоящего оборудования;
- требование наличия только компьютера и мультимедийного проектора;
- возможность быстро изменять параметры в широких пределах;
- возможность модельного эксперимента, показывающего любое явление в отличие от натурной демонстрации.

Необходимо различать термины “компьютерная модельная демонстрация” и “демонстрационная программа”. Первая является средством наглядности, используемым в учебном процессе. Вторая же является одним из технических средств, с помощью которых эта демонстрация реализуется. Демонстрация имеет определённую методику показа; одна и та же программа может лежать в основе нескольких различных демонстраций, использующих разные ее возможности. Преимущества модельной демонстрации вполне могут компенсироваться недостатками компьютерной программы.

Можно выделить ряд типичных недостатков программ, полученных из анализа уже имеющихся на сегодняшний день компьютерных демонстраций:

- малый размер окна программы;
- неудачный выбор размера или начертания шрифта;
- низкое качество изображения (по разрешению или цветам);
- сложный выбор значения параметров;
- избыточное количество параметров;
- неудачное расположение элементов интерфейса и т.п.

Требования, которым должна удовлетворять любая модельная программа, вытекают из требований к компьютерным демонстрациям, реализуемым с их помощью.

Их можно разделить на несколько групп:

- дидактические;
- технические;
- психофизиологические;
- эргономические.

## 1.1 Дидактические и технические требования к модельным программам

*Демонстрационные программы* входят в более широкий класс *лекционных демонстраций (лекционных экспериментов)*. В методике преподавания физики сформулирован ряд дидактических требований к ним [8]. Рассмотрим как они могут быть реализованы в демонстрационной программе:

- **наглядность:** наглядность результатов работы программы зависит от многих факторов, например, корректное воспроизведение изображения и правильное представление числовой информации [7];
- **выразительность:** необходимость отражения посредством демонстрационной программы главного и наиболее общего в модели, используемой для объяснения явления;
- **убедительность:** у зрителей (учащихся) не должно возникать возможности двойственно или неправильно истолковать увиденное. Необходимы простота и чёткость построения схемы интерфейса программы, определяющие простоту интерпретации увиденного;
- **надежность:** возможность повторного исполнения программы;
- **кратковременность:** соответствие темпа проведения демонстрации темпу изложения, отсутствие длительных пауз — например, для перенастройки параметров;
- **занимательность:** демонстрация модельной программы должна вызывать интерес, поддерживать внимание зрителей к ней. Для этого интерфейс программы должен иметь цветояркие акценты, содержать элементы анимации;
- **видимость:** все важные для интерпретации учебного материала детали моделируемой установки, а также манипуляции с ней должны быть хорошо видны и понятны зрителям;
- **безопасность:** программа не должна создавать изображений, опасных для их наблюдения зрителями с нарушениями нервной деятельности.  
Также необходимо отметить важные технические требования:
- **интерактивность:** возможность изменения условий проведения модельного эксперимента;
- **надежность:** должна быть предусмотрена внутренняя проверка данных, предотвращающая введение некорректных настроек и аварийное прекращение работы программы;
- **переносимость:** возможность исполнения на разных компьютерах без проведения специальных настроек операционной системы; в идеале — кроссплатформ-

менность (интероперабельность), то есть возможность работы программ под управлением различных операционных систем.

Многие требования (такие, как наглядность, выразительность, убедительность, кратковременность, видимость) на первый взгляд могут показаться нечеткими. Ведь простота, доступность и т.п. могут быть отнесены к субъективным факторам. Однако у них существуют и общие, эргономические закономерности, основанные на психофизиологических факторах.



## 1.2 Психофизиологические и эргономические требования к модельным программам

Одним из характерных отличий демонстрационной программы, используемой на лекции, от любой другой компьютерной программы является очень малое время взаимодействия с ней аудитории. При демонстрации на лекции какого-либо эксперимента или просто изложения материала важно, чтобы внимание зрителей концентрировалось там, где это требуется в данный момент. Для этого лектор специально акцентирует внимание слушателей на том, что считает важным. Демонстрационные экспериментальные установки проектируются, собираются и устанавливаются с тем расчетом, чтобы в нужный момент привлечь внимание зрителей к физическому явлению.

При работе на лекции с демонстрационной программой также необходимо учитывать факторы, обеспечивающие привлечение внимания. Однако при разработке программы они, как правило, игнорируются или учитываются не вполне корректно. То, что представляется хорошо видимым на экране монитора при индивидуальной работе с ЭВМ, может оказаться малозаметным при показе в аудитории с использованием вспомогательных технических средств (например, мультимедийных проекторов), с другого, причём фиксированного, ракурса, при отличающемся освещении. Внимание аудитории будет ослабевать из-за того, что возрастут непроизводительные затраты, связанные с напряжением зрения и необходимостью распознавания увиденного.

Если возникают проблемы при работе с программой, то она создает паузы в лекции и заставляет отвлекаться от учебного процесса, ослабляет внимание аудитории или концентрирует его на посторонних вещах. Безусловно, эти проблемы не являются непреодолимыми, и их наличие вряд ли сделает невозможным использование программы. Однако их учет позволит существенно повысить внимание студентов непосредственно к физическим явлениям, моделируемым демонстрационной программой, и увеличить эффективность работы лектора с аудиторией. Все это может оказать влияние на повышение эффективности лекции в целом.

**Внимание** можно определить как “сосредоточенность деятельности субъекта в данный момент времени на каком-либо реальном или идеальном объекте (предмете, событии, образе, рассуждении и т. д.)” [10]. Состояние внимания противоположно рассеянному состоянию. Внимание необходимо человеку для понимания происходящего, точной оценки ситуации и анализа сложных систем, понимание которых занимает много времени. Выделяют три вида внимания [5].

**Произвольное**, или активное, внимание характеризуется направленностью субъекта на сознательно выбранную цель. Однако существует форма и **непроиз-**

**вольного**, или пассивного внимания. Оно выражается в переключении внимания на неожиданное изменение или появление значимых сигналов. Третий вид внимания называется **постпроизвольным** вниманием, которое появляется в процессе освоения деятельности и увлеченности выполняемой работой. Оно не требует усилий воли, так как поддерживается интересом к ней. К характеристикам внимания относят его избирательность, объем, устойчивость, возможность распределения и переключения.

**Избирательность** внимания характеризует его направленность на любой аспект стимула. Избирательность внимания измеряется количеством одновременно отчетливо осознаваемых объектов и характеризует ресурсы внимания человека. **Объем** внимания близок объему кратковременной памяти и составляет 7-9 элементов. **Устойчивость** внимания определяется по длительности выполнения задания, требующего непрерывного внимания. Под **переключением** внимания понимается возможность более или менее легкого перехода от одного вида деятельности к другому.

Чтобы учитывать особенности внимания учащихся, сформулируем на основе обобщения данных из научно-методической литературы [7, 5]. Основные психофизиологические требования к интерфейсу демонстрационных программ:

- наиболее важные в данный момент элементы программы должны быть лучше всего видны;
- моделируемое изображение должно быть хорошо видно, то есть иметь большой размер, высокие яркость и контрастность;
- все существенные элементы должны быть видимы одновременно;
- одновременно в поле зрения должно быть не более 7-9 элементов;
- вспомогательные элементы не должны быть в поле зрения;
- внешний вид программы должен учитывать особенности непроизвольного внимания учащихся;
- должно быть выбрано цветовое содержание (например, цветовое зонирование), повышающего концентрацию внимания на определенных элементах интерфейса.

Эргономика — научно-прикладная дисциплина, занимающаяся изучением и созданием эффективных систем, управляемых человеком [10]. Эргономика изучает движение человека в процессе производственной деятельности, затраты его энергии, производительность и интенсивность при конкретных видах работ.

Эргономические требования к интерфейсу демонстрационной программы могут быть сформулированы на основе требований к пользовательскому интерфейсу [7]:

- простота установки начальных параметров для программ и изменения их значений в процессе работы;

- интерактивность — возможность изменения условий проведения моделирования.

При повторной работе с программой возникают дополнительные критерии, например скорость восстановления навыка работы с интерфейсом и возможность возврата к ранее использовавшимся настройкам программы.

Сформулировав все необходимые требования к демонстрационным программам, необходимо перейти к разработке общей концепции построения интерфейсов, реализующих эти требования.

## 1.3 Операторский и демонстрационный интерфейсы

Демонстрационный интерфейс подразумевает восприятие информации без ее ввода.

Интерфейс — определенная граница между взаимодействующими независимыми объектами. При взаимодействии пользователя с компьютерной программой мы рассматриваем их как два независимых объекта. При использовании демонстрационных программ на лекции необходимо рассматривать не два, а три независимых объекта. Это лектор, зрители (слушатели лекции) и демонстрационная программа. Подобное разделение обусловлено различием информационных потоков, возникающих на лекции во время показа модельных демонстраций.

Интерфейс, определяющий в условиях проведения лекции границу взаимодействия между программой и лектором, назовем операторским. Операторский интерфейс подразумевает восприятие не только с монитора персонального компьютера, но и с другого оборудования для вывода информации (проекционный телевизор, мультимедийный проектор). Информативность операторского интерфейса должна быть направлена больше на зрителя, нежели чем на самого оператора.

Интерфейс, определяющий в условиях проведения лекции границу взаимодействия между программой и зрителями, назовем демонстрационным.

## 1.4 Концепция интерфейса модельных демонстраций

Высокоуровневые языки программирования всегда предоставляют в пользование разработчиков программного обеспечения определенный базовый набор интерфейсных элементов. Эти элементы проектируются с тем расчетом, чтобы их возможности позволяли решать большинство часто встречающихся интерфейсных задач.

Но при использовании базового набора интерфейсных элементов для написания интерфейса необходимо понимать, что демонстрационные программы представляют собой специфическое направление компьютерного моделирования. Анализ существующего набора интерфейсов демонстрационных программ по оптике показывает, что для правильной работы таких программ необходимо наличие всего нескольких основных функций интерфейса [7]:

- ввод/вывод числовой информации;
- функция активации/остановки работы программы;
- вывод наглядных результатов моделирования;
- вывод количественных результатов моделирования (графиков);
- отображение схемы установки;

- отображение и настройка спектра источников света.

Реализация этих функций с использованием доступных интерфейсам базовых элементов позволяет решать все поставленные задачи.

Однако интерфейс, составленный из стандартных для визуальной среды базовых интерфейсных элементов, представленных в среде разработки по умолчанию, не ориентирован на пассивное восприятие информации зрителем. Поэтому необходимо разработать иные базовые элементы, как на основе имеющихся, так и на основе использования или сочетания принципиально новых решений.

## Глава 2

# Принципы построения и типовые элементы интерфейса демонстрационных модельных программ

Опираясь на требования, сформулированные в первой главе, для реализации модельных программ был выбран язык программирования C++ с использованием высокоуровневой библиотеки Qt. Причины этого следующие:

- использование объектно-ориентированного языка программирования дает мощьность и гибкость в решении задач, связанных со структурным разбиением модели на отдельные элементы данных;
- необходимость визуализации модели требует реализации программы в графической среде, для чего в C++ существует ряд программных библиотек. Графические среды предоставляются всеми современными операционными системами;
- на данный момент для C++ уже существует большое количество библиотек, реализующих различные численные методы или физические модели. Это может упростить процесс разработки новых программ.

В связи с необходимостью визуализации реализация в графической среде. Такие среды предоставляются рядом операционных систем. Реализация кроссплатформенности, то есть поддержка различных операционных систем, существует в ряде библиотек, например, Qt, Gd, Delphi 7/Kylix. Из них только Qt работает с C++ и имеет объектно-ориентированную архитектуру.

Для реализации наглядных, реалистичных, быстро генерирующихся изображений эта библиотека позволяет использовать библиотеку OpenGL.

## 2.1 Библиотека QT

Для создания программ была выбрана библиотека QT [2, 12]. Trolltech Qt — кросс-платформенный инструментарий разработки программного обеспечения на языке программирования C++, имеющий «привязки» ко многим другим языкам программирования.

Библиотека Qt позволяет запускать написанное с ее помощью программное обеспечение в большинстве современных операционных систем (ОС семейств Microsoft Windows, Linux, Apple MacOS) путем простой перекомпиляции программы для каждой ОС без изменения исходного кода. Она включает в себя все основные классы, которые могут потребоваться при разработке прикладного программного обеспечения, начиная от элементов графического интерфейса и заканчивая классами для работы с сетью, базами данных и XML. Qt является полностью объектно-ориентированной, легко расширяемой и поддерживающей технику компонентного программирования. Таким образом, использование этой библиотеки позволяет создать учебно-методическую среду, одинаковую при работе в различных операционных системах: Microsoft Windows, Linux/UNIX, MacOS.

Библиотека Qt является свободно распространяемой (согласно лицензиям GPL-2 или QPL-1, с определенными ограничениями в случае коммерческого использования), что становится актуальным в связи с лицензированием программного обеспечения, используемого в учебном процессе.

Со времени своего появления в 1996 году коммерческая версия библиотеки Qt легла в основу тысяч успешных проектов во всем мире. Кроме того, Qt является фундаментом популярной рабочей среды KDE, входящей в состав многих дистрибутивов GNU/Linux.

Перечислим основные достоинства Qt, определившие наш выбор этой библиотеки для создания программы.

1. Объектно-ориентированное программирование — это парадигма программирования, в которой основными концепциями являются понятия объектов и классов. Класс — это тип, описывающий устройство объектов-экземпляров, инкапсулирующий (т.е. включающий в себя) как данные, так и процедуры их обработки. Возможно как наследование данных и процедур, так и их полиморфизм (изменение). Объектно-ориентированное программирование возникло в результате развития идеологии процедурного программирования, где данные и процедуры обработки формально не связаны. Объектно-ориентированное программирование в настоящее время является абсолютным лидером в области прикладного программирования. В то же время в области системного программирования до

сих пор лидирует сугубо процедурный язык С, хотя при взаимодействии системного и прикладного уровней операционных систем заметное влияние стали оказывать языки объектно-ориентированного программирования. Поэтому Qt, написанная на языке С++, стала своеобразным мультиплатформенным стандартом.

2. Большое число стандартных функции, необходимых для численного моделирования. Так, например, в библиотеке уже определены классы векторов, матриц и операции над ними.
3. Мощные и удобные средства для разработки графического пользовательского интерфейса, включающие в себя все необходимые стандартные элементы, которые к тому же могут быть легко модифицированы для конкретной программы.
4. Интеграция в различные среды разработки программного обеспечения, как коммерческие (Microsoft Visual Studio 2003/2005), так и свободно распространяемые (Dev-Cpp, CodeBlocks и т.п.).
5. Простота и гибкость программирования, основанная на концепции слотов и сигналов, повышающая эффективность и скорость разработки программы.

Сигналы и слоты используются для связи между объектами Qt. Механизм сигналов и слотов является важнейшей возможностью Qt и, возможно, тем, чем Qt больше всего отличается от остальных инструментариев, часто использующих использующих механизмы обратной связи. Сигналы исходят от объектов, когда происходит какое-то событие (например, пользователь печатает что-то в поле ввода или нажимает кнопку). Слот — это функция, вызываемая в ответ на определенный сигнал. Средства Qt позволяют просто связать сигнал и соответствующий ему слот.



## 2.2 Библиотека OpenGL

OpenGL (Open Graphics Library — открытая графическая библиотека) — спецификация, определяющая независимый от языка программирования кросс-платформенный программный интерфейс для написания приложений, использующих двумерную и трёхмерную компьютерную графику. Включает в себя более 250-ти функций для рисования сложных трёхмерных сцен из простых примитивов.

На базовом уровне, OpenGL — это просто спецификация, то есть документ, описывающий набор функций и их точное поведение. Производители оборудования на основе этой спецификации создают реализации — библиотеки функций, соответствующих набору функций спецификации. Реализация использует возможности оборудования там, где это возможно. Если аппарататура не позволяет реализовать какую-либо возможность, она должна быть эмулирована программно. Производители должны пройти специфические тесты (conformance tests — тесты на соответствие) прежде чем реализация будет классифицирована как OpenGL реализация. Таким образом, разработчикам программного обеспечения достаточно научиться использовать функции, описанные в спецификации, оставив эффективную реализацию последних разработчикам аппаратного обеспечения.

Эффективные реализации OpenGL существуют для Windows, Unix-платформ и Mac OS. Основным принципом работы OpenGL является получение наборов векторных графических примитивов в виде точек, линий и многоугольников с последующей математической обработкой полученных данных и построением растровой картинке на экране и/или в памяти. Векторные трансформации и растеризация выполняются графическим конвейером (graphics pipeline), который по сути представляет из себя дискретный автомат. Абсолютное большинство команд OpenGL попадают в одну из двух групп: либо они добавляют графические примитивы на вход в конвейер, либо конфигурируют конвейер на различное исполнение трансформаций.

OpenGL является низкоуровневым процедурным API, что вынуждает программиста диктовать точную последовательность шагов, чтобы построить результирующую растровую графику (императивный подход). Это является основным отличием от дескрипторных подходов, когда вся сцена передается в виде структуры данных (чаще всего дерева), которое обрабатывается и строится на экране. С одной стороны, императивный подход требует от программиста глубокого знания законов трёхмерной графики и математических моделей, с другой стороны — даёт свободу внедрения различных инноваций.

## 2.3 Структуры, потоки данных и элементы интерфейса

Анализ модельных программ позволил разбить их на ряд структур, обменивающихся потоками данных. Их можно разбить на три типа:

1. Внутренняя структура модели: собственно физическая модель, то есть, например, формула для разности хода для модели интерферометра и т.п.
2. Внешние структуры (параметры) модели:
  - спектральный состав излучения — используется во всех моделях;
  - параметры установки (например, показатель преломления какого-либо материала, коэффициент отражения, радиус отверстия и т.п.) — их количество и состав специфичны для каждой программы.
3. Структуры отображения:
  - количественное отображение информации подразумевает построение графиков (только условно-символический вид);
  - качественное отображение информации представляет ее в наглядном или условно-символическом виде.

Такое разбиение позволяет определить потоки данных. Они направлены от объектов, расположенных в списке выше, к более низкорасположенным объектам. Например, изменение модели приводит к изменению количества и состава ее параметров, а также требует обновления всех структур отображения. Изменение же значения какого-либо параметра не затрагивает структуру модели, а должно приводить лишь к новому отображению.

Пользователь взаимодействует с перечисленными структурами через отдельные элементы интерфейса программы — виджеты. Они могут иметь иерархическую структуру, включая в себя другие элементы интерфейса (также виджеты). Выделим основные виджеты, из которых будет состоять интерфейс любой программы:

- виджет управления параметрами спектрального состава излучения (тип 2);
- виджет управления параметрами установки (тип 2);
- виджет отображения схемы установки (тип 3);
- виджет наглядного результата моделирования (тип 3);
- виджет количественного результата моделирования (тип 3);
- виджет пользовательского интерфейса, объединяющий все вышеперечисленные виджеты.

Поскольку задание значений различных параметров с помощью стандартных элементов управления не всегда удовлетворяет эргономическим требованиям, был

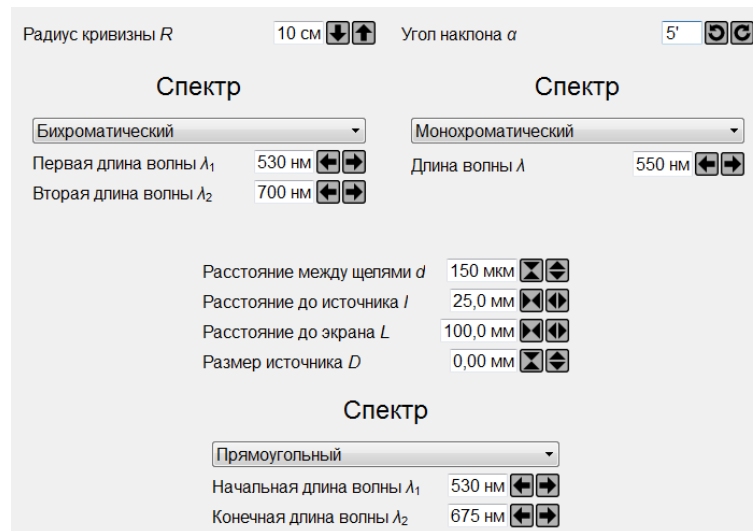


Рис. 2.1: Внешний вид виджетов задания параметров установки и спектра

создан отдельный виджет более низкого уровня. Он принимает различный внешний вид при управлении разными величинами, связывая изменение параметра с визуальными изменениями во внешнем виде установки.

Этот виджет используется в более сложных: управления параметрами установки и управления спектром излучения. Виджет управления спектром излучения имеет как регулируемые режимы (например, у монохроматического излучения можно выбрать длину волны), так и предустановленные (например, излучение гелий-неонового лазера).

## 2.4 Отображение цвета монохроматического излучения

Проблемы воспроизведения цвета немонохроматического излучения сложного спектрального состава подробно рассмотрены в [8]. Однако часто требуется отображение цвета монохроматического излучения. Для такого отображения можно использовать более простой (а, значит, и более быстрый) алгоритм.

Формулы для расчета  $RGB$ -компонентов цвета для нужной длины волны приведены в табл. 2.1. Показатель нелинейности монитора  $\gamma$  был выбран равным 1.8, что соответствует большинству моделей мониторов. В таблице записаны компоненты цвета, соответствующие максимальной яркости изображения. Для ослабления яркости в  $n$  раз нужно разделить полученное значение на  $n^\gamma$ .

$\lambda$	$R(\lambda)$	$G(\lambda)$	$B(\lambda)$
380..420	$\left(\frac{440 - \lambda}{440 - 380}\right)^\gamma \cdot \left(0.3 + 0.7 \frac{\lambda - 380}{420 - 380}\right)^\gamma$	0	$\left(0.3 + 0.7 \frac{\lambda - 380}{420 - 380}\right)^\gamma$
420..440	$\left(\frac{440 - \lambda}{440 - 380}\right)^\gamma$	0	1
440..490	0	$\left(\frac{\lambda - 440}{490 - 440}\right)^\gamma$	1
490..510	0	1	$\left(\frac{510 - \lambda}{510 - 490}\right)^\gamma$
510..580	$\left(\frac{\lambda - 510}{580 - 510}\right)^\gamma$	1	0
580..645	1	$\left(\frac{645 - \lambda}{645 - 580}\right)^\gamma$	0
645..700	1	0	0
700..780	$\left(0.3 + 0.7 \frac{780 - \lambda}{780 - 700}\right)^\gamma$	0	0

Таблица 2.1: К расчету цвета монохроматического излучения:  $\lambda$  — длина волны излучения в нанометрах,  $R$ ,  $G$ ,  $B$  — красный, зеленый и синий компоненты соответственно,  $\gamma$  — показатель нелинейности монитора.

## Глава 3

# Созданные демонстрационные программы

### 3.1 Программы по интерференции света

Эти программы содержат в себе все перечисленные во второй главе элементы интерфейса. Для удобства использования они скомпонованы в единую программу.

#### 3.1.1 Интерферометр Майкельсона

Эта программа позволяет моделировать и наблюдать интерференционную картину в равноплечном интерферометре Майкельсона при освещении его параллельным пучком света, а также производить расчет и построение графика зависимости интенсивности от поперечной координаты с использованием различных источников излучения (монохроматических и немонахроматических).

Если зеркала не строго перпендикулярны, то интерференционная картина будет аналогична картине, наблюдаемой в тонком клине. Разность хода между лучами в точке наблюдения, имеющей координату  $x$ , равна 3.1:

$$\Delta = 2(x \operatorname{tg} \alpha) \quad (3.1)$$

где  $\alpha$  — угол отклонения зеркал от взаимной перпендикулярности. Интенсивность наблюдаемой картины в монохроматическом свете будет находиться из соотношения:

$$I = I_0 (1 + \cos[k\Delta]).$$

Интерфейс программы изображен на рисунке 3.1. Он разделен на два столбца. В левом расположены (сверху вниз) схема моделируемой установки, элементы управления ее параметрами, а также виджет управления спектром излучения; в правом — наглядный и количественный результаты моделирования. В роли количественного результата выступает график зависимости интенсивности изображения от попе-

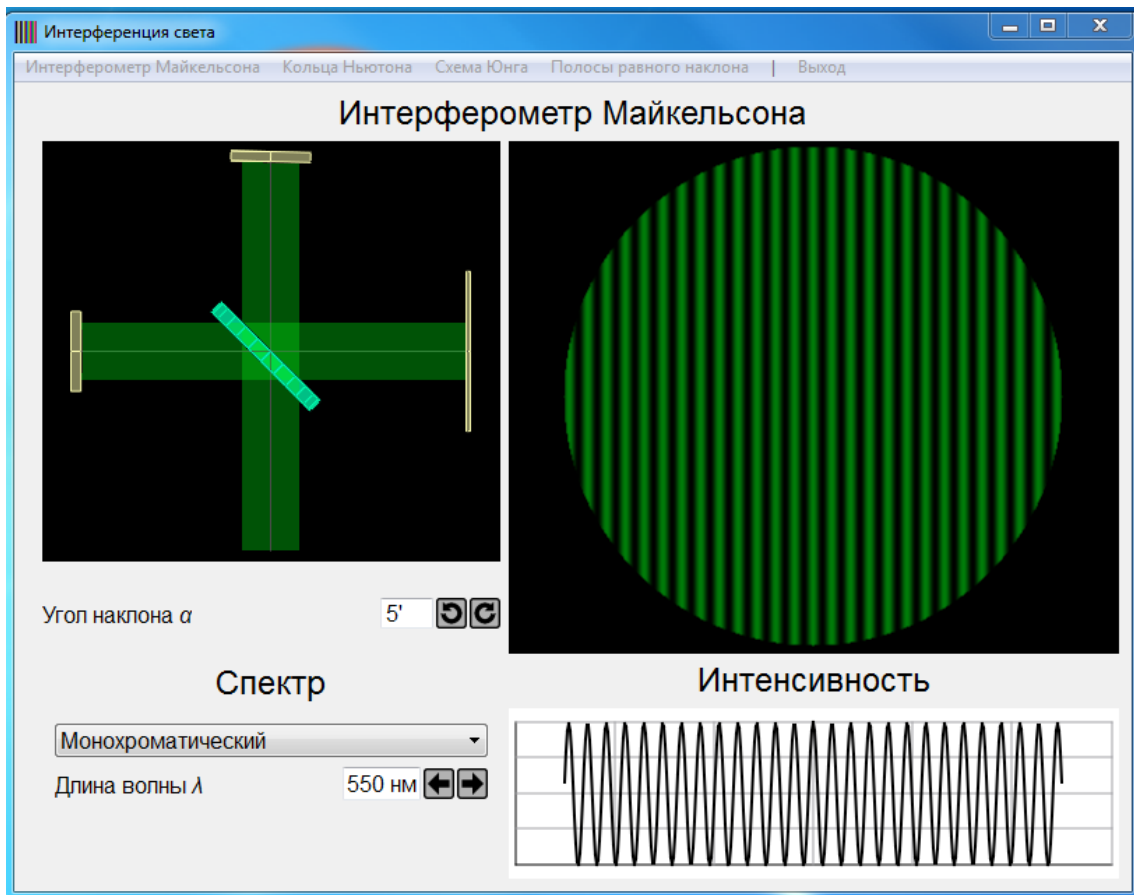


Рис. 3.1: Интерфейс модельной программы "Интерферометр Майкельсона"

речной координаты, пространственно согласованный с наглядным результатом для возможности их сравнения.

Программа позволяет наблюдать следующие закономерности интерференционных картин:

- в монохроматическом свете: изменение вида картины (частоты полос) в зависимости от длины волны;
- в бихроматическом свете: возникновение “смазываний” картины в зависимости от разности длин волн;
- в прямоугольном спектре; зависимость “смазываний” от ширины спектра, сравнение с случаем бихроматического источника;
- в свете гелий–неонового лазера: сравнение с картиной, наблюдаемой в натурном эксперименте.

### 3.1.2 Кольца Ньютона

Эта программа позволяет моделировать и наблюдать интерференционную картину в схеме с делением амплитуды (кольца Ньютона) в системе из плоской и выпуклой стеклянных пластин при наблюдении в отраженном свете, а также производить рас-

чет и построение графика зависимости интенсивности от радиальной координаты с использованием различных источников излучения (монохроматических и немонахроматических).

Падающий коллимированный пучок света отражается от системы, состоящей из двух соприкасающихся пластин, выпуклой и плоской. Интерференционная картина формируется пучками, отраженными от обращенных друг к другу поверхностей пластин. Интенсивность интерференционной картины рассчитывается следующим образом. В общем случае разность хода  $\Delta$  между интерферирующими волнами в точке с координатой  $\rho$  равна 3.2:

$$\Delta(\rho) = \frac{\rho^2}{2} \cdot \frac{1}{r}, \quad (3.2)$$

где  $r$  — радиус кривизны первой пластины,  $\rho$  — расстояние от рассматриваемой точки до точки соприкосновения пластин (радиальная координата).

Считая интенсивности интерферирующих отраженных пучков одинаковыми (что соответствует практически равным коэффициентам отражения света от поверхностей обеих пластин), в монохроматическом свете с длиной волны  $\lambda$  радиальное распределение интенсивности представляется в виде

$$I(\rho) = I_0 \cdot \left( 1 + \cos \left[ \frac{2\pi\Delta(\rho)}{\lambda} + \pi \right] \right),$$

а для источника со спектральной плотностью  $I(\lambda)$  в интервале длин волн от  $\lambda_{min}$  до  $\lambda_{max}$

$$I(\rho) = \int_{\lambda_{min}}^{\lambda_{max}} I(\lambda) \left( 1 + \cos \left[ \frac{2\pi\Delta(\rho)}{\lambda} + \pi \right] \right) d\lambda,$$

Принято, что коэффициент отражения не зависит от длины волны падающего света.

Если источник бихроматический и длины волн излучения равны  $\lambda_1$  и  $\lambda_2$ , то интенсивность в точке наблюдения с координатой  $\rho$  будет описываться соотношением:

$$I = 2I_0 \left( 1 + \cos \left( \frac{k_1 + k_2}{2} \Delta(\rho) \right) \cos \left( \frac{k_1 - k_2}{2} \Delta(\rho) \right) \right),$$

Если источник имеет прямоугольный спектр с длинами волн от  $\lambda_1$  до  $\lambda_2$ , то интенсивность в точке наблюдения будет описываться соотношением:

$$I = 2I_0 (1 + \text{sinc}((k_2 - k_1)\Delta) \cos(k_0\Delta)).$$

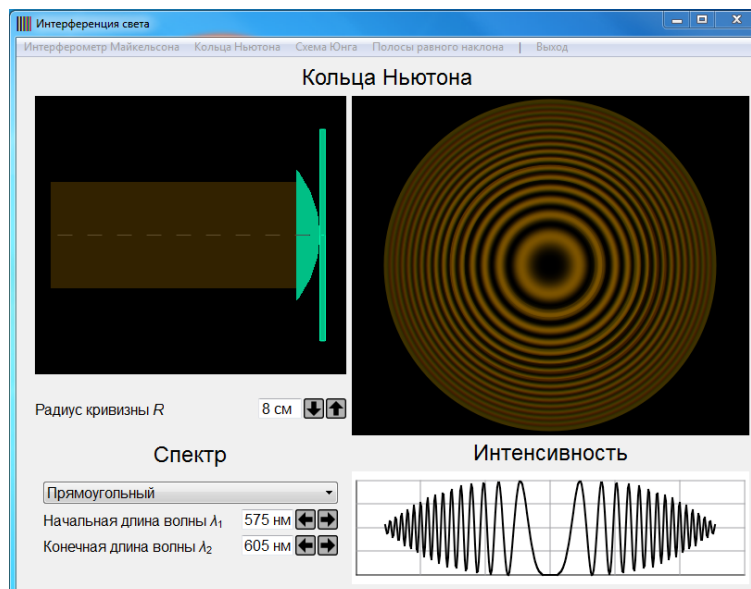


Рис. 3.2: Интерфейс модельной программы "Кольца Ньютона"

Интерфейс программы изображен на рисунке 3.2. Он разделен на два столбца. В левом расположены (сверху вниз) схема моделируемой установки, элементы управления ее параметрами, а также виджет управления спектром излучения; в правом — наглядный и количественный результаты моделирования. В роли количественного результата выступает график зависимости интенсивности изображения от поперечной координаты, пространственно согласованный с наглядным результатом для возможности их сравнения.

Программа позволяет наблюдать интерференционную картину:

- в монохроматическом свете: изменение вида картины (частоты радиуса колец) в зависимости от длины волны;
- в бихроматическом свете: возникновение “смазываний” картины в зависимости от разности длин волн;
- в прямоугольном спектре; зависимость “смазываний” от ширины спектра, сравнение с случаем бихроматического источника;
- в свете гелий–неонового лазера: для сравнения с картиной, наблюдаемой в натурном эксперименте.

### 3.1.3 Схема Юнга

Эта программа позволяет моделировать и наблюдать интерференционную картину в схеме Юнга, а также производить расчет и построение графика зависимости интенсивности от радиальной координаты с использованием различных источников излучения (монохроматических/немонохроматических, точечных/протяженных).



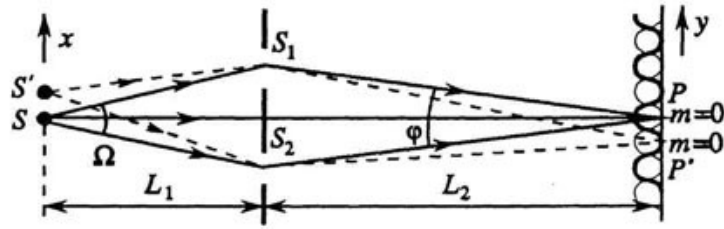


Рис. 3.3: К расчету интерференционной картины в схеме Юнга для двух точечных источников

Интерференционная схема Юнга состоит из источника (в качестве которого выступает узкая щель на экране), экрана с двумя близко расположенными щелями, и экрана, на котором наблюдается интерференционная картина.

Разность хода между двумя лучами, приходящими в одну точку экрана от двух щелей, находится по формуле:

$$\Delta = d \sin(\phi) \approx \frac{dx}{L},$$

где  $d$  – расстояние между щелями,  $\phi$  – угол отклонения лучей от направления прямолинейного распространения,  $x$  – координата точки наблюдения на экране,  $L$  – расстояние между экраном с щелями и экраном наблюдения.

Интенсивность в точке наблюдения с координатой  $x$  описывается соотношением:

$$I = 2I_0 (1 + \cos(k\Delta))$$

Если источник бихроматичный, и длины волн излучения равны  $\lambda_1$  и  $\lambda_2$ , то интенсивность в точке наблюдения с координатой  $X$  будет описываться соотношением:

$$I = 2I_0 \left( 1 + \cos\left(\frac{k_1 + k_2}{2}\Delta\right) \cos\left(\frac{k_1 - k_2}{2}\Delta\right) \right)$$

Если источник имеет прямоугольный спектр с длинами волн от  $\lambda_1$  до  $\lambda_2$ , то интенсивность в точке наблюдения будет описываться соотношением:

$$I = 2I_0 (1 + \text{sinc}((k_2 - k_1)\Delta) \cos(k_0\Delta))$$

Для случая двух источников (см. рис. 3.3), важно обратить внимание, что при смещении источника на расстояние  $x$ , интерференционная картина сместится вдоль оси на расстояние  $y = -\left(\frac{L_2}{L_1}\right)x$ , складывая по интенсивности интерференционные картины от каждого из источников, можно получить результирующую интерференционную картину. Протяженный источник разбивается на источники размера много

меньшего, чем длина протяженного источника.

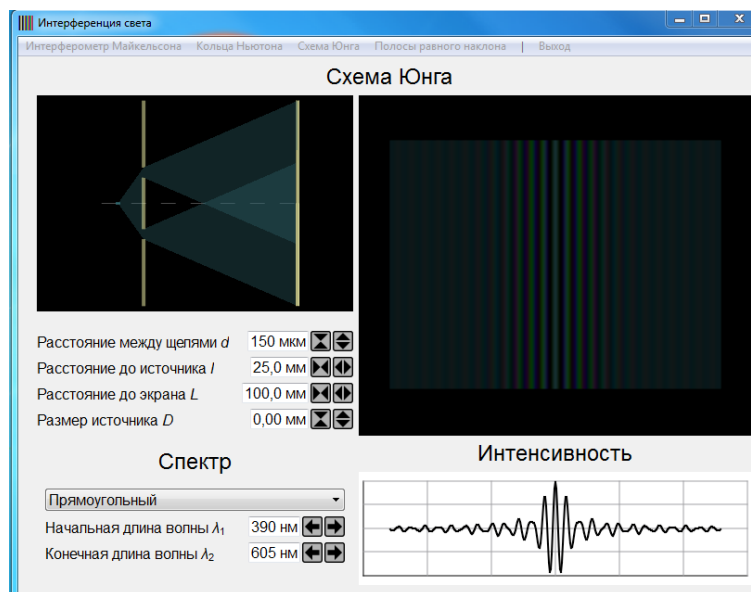


Рис. 3.4: Интерфейс модельной демонстрации "Схема Юнга"

Интерфейс программы изображен на рисунке 3.4. Он разделен на два столбца. В левом расположены (сверху вниз) схема моделируемой установки, элементы управления ее параметрами, а также виджет управления спектром излучения; в правом — наглядный и количественный результаты моделирования. В роли количественного результата выступает график зависимости интенсивности изображения от поперечной координаты, пространственно согласованный с наглядным результатом для возможности их сравнения.

Программа позволяет наблюдать интерференционную картину:

- в монохроматическом свете: изменение вида картины (частоты полос) в зависимости от длины волны;
- в бихроматическом свете: возникновение “смазываний” картины в зависимости от разности длин волн;
- в прямоугольном спектре; зависимость “смазываний” от ширины спектра, сравнение с случаем бихроматического источника;
- при освещении протяженным источником: изменение видности интерференционной картины при изменении размера источника;
- в свете гелий–неонового лазера: для сравнения с картиной, наблюдаемой в натурном эксперименте.

### 3.1.4 Тонкая пластинка (полосы равного наклона)

Эта программа позволяет моделировать и наблюдать интерференционную картину в тонкой плоскопараллельной пластине, а также производить расчет и построение графика зависимости интенсивности от радиальной координаты с использованием различных источников излучения (монохроматических и немонахроматических).

Разность хода интерферирующих лучей находится по формуле

$$\Delta = 2nh \cos \beta,$$

или

$$\Delta = 2nh \sqrt{1 - \frac{x^2}{n^2(x^2 + f^2)}}$$

Учитывая, что при отражении волны от границы воздух-стекло, ее фаза меняется на  $\pi$ , можем записать выражение для интенсивности интерференционной картины в виде:

$$I = I_0 \left( 1 + \cos \left[ k \left( \Delta + \frac{\lambda}{2} \right) \right] \right),$$

или

$$I = I_0 \left( 1 + \cos \left[ \frac{4\pi h \sqrt{n^2 - \sin^2 \alpha}}{\lambda \pm \pi} \right] \right)$$

В случае немонахроматического источника, спектр которого содержит две длины волны  $\lambda_1$  и  $\lambda_2$ , выражение для интенсивности интерференционной картины примет вид:

$$I = I_0 \left( 1 + \cos \left[ \frac{\Delta(k_1 + k_2)}{2} + \pi \right] \cos \left[ \frac{\Delta(k_1 - k_2)}{2} \right] \right)$$

а для источника с прямоугольным спектром от  $\lambda_1$  до  $\lambda_2$ :

$$I = 2I_0(1 + \text{sinc}((k_2 - k_1)\Delta) \cos(k_0\Delta))$$

Интерфейс программы изображен на рисунке 3.5. Он разделен на два столбца. В левом расположены (сверху вниз) схема моделируемой установки, элементы управления ее параметрами, а также виджет управления спектром излучения; в правом — наглядный и количественный результаты моделирования. В роли количественного результата выступает график зависимости интенсивности изображения от поперечной координаты, пространственно согласованный с наглядным результатом для возможности их сравнения.

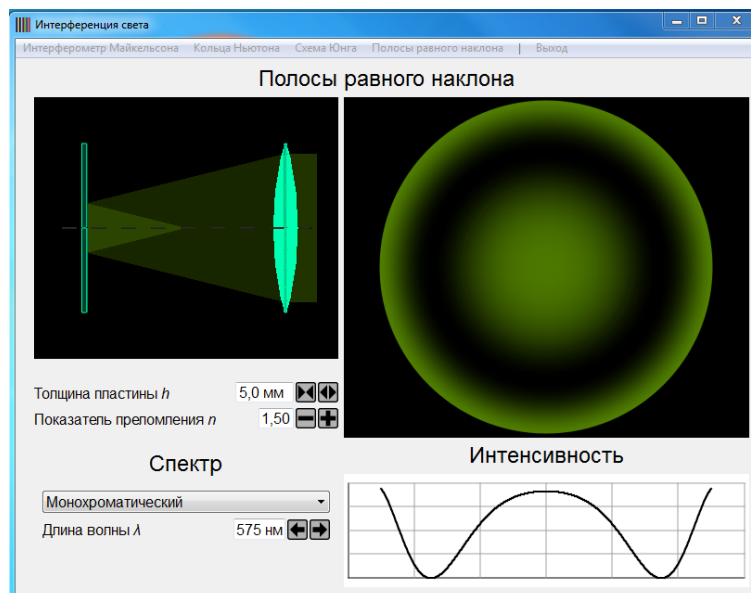


Рис. 3.5: Модельная демонстрация полос равного наклона

Программа позволяет наблюдать интерференционную картину:

- в монохроматическом свете: изменение вида картины (радиусы колец) в зависимости от длины волны;
- в бихроматическом свете: возникновение “смазываний” картины в зависимости от разности длин волн;
- в прямоугольном спектре; зависимость “смазываний” от ширины спектра, сравнение с случаем бихроматического источника;
- в свете гелий–неонового лазера: для сравнения с картиной, наблюдаемой в натурном эксперименте.

## 3.2 Программы по дифракции света

### 3.2.1 Дифракция Френеля на круглом отверстии

#### Зоны Френеля

Вместо точного вычисления интеграла Френель предложил приближенный способ расчета дифракционных картин, основанный на представлении о полуволновых зонах (зонах Френеля). Они вводятся следующим образом.

Выберем поверхность  $\Sigma$  в виде сферы с центром в точке  $S$  (рис. 3.6). Согласно принципу Гюйгенса–Френеля, эту поверхность можно рассматривать как источник вторичных световых волн. Выделим на сфере кольцевые зоны так, чтобы расстояния от границ зон до точки наблюдения отличались на половину длины световой волны. Обозначив границы зон буквами  $M_0, M_1, M_2, \dots$ , получим:

$$\begin{cases} M_0P = OP + \lambda/2 \\ M_1P = M_0P + \lambda/2 \\ \dots\dots\dots \\ M_nP = M_{n-1}P + \lambda/2, \end{cases} \quad (3.3)$$

где  $\lambda$  — длина световой волны,  $P$  — точка наблюдения поля,  $O$  — центр нулевой зоны Френеля.

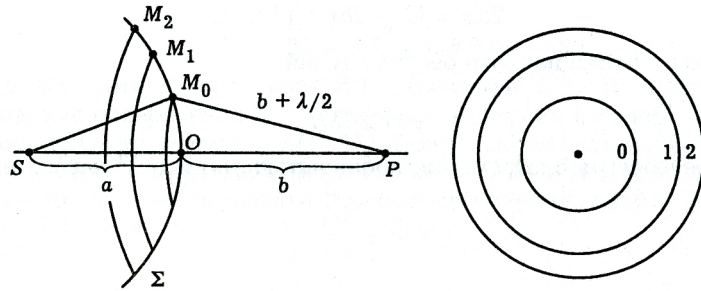


Рис. 3.6: Зоны Френеля.

Формулы (3.3) определяют положение границ зон Френеля. Смысл разбиения поверхности  $\Sigma$  на зоны состоит в том, что разность фаз любых вторичных волн, приходящих в точку наблюдения от данной зоны, не превышает  $\pi$ . Сложение таких волн приводит к их взаимному усилению. Поэтому каждую зону Френеля можно рассматривать как источник вторичных волн, имеющих определённую фазу. Напротив, две соседние зоны Френеля действуют как источники, колеблющиеся в противофазе. Положение границ френелевских зон зависит от выбора точки наблюдения.

Для того, чтобы оценить относительный вклад френелевских зон в интеграл Гюйгенса–Френеля, оценим радиусы зон и их площади.

На рис. 3.7 показаны точечный источник света  $S$ , точка наблюдения поля  $P$ , часть сферической поверхности  $\sigma$  (источника вторичных волн) и граница нулевой зоны Френеля  $M_0$ . Пусть  $a$  — радиус сферы,  $b$  — кратчайшее расстояние от точки  $P$

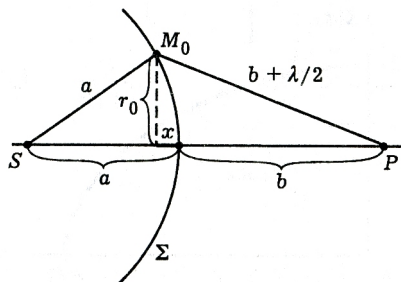


Рис. 3.7: К расчету размеров френелевских зон.

до сферы,  $r_0$  — радиус нулевой зоны Френеля. Из рис. 3.7 видно, что

$$r_0^2 = a^2 - (a - x)^2, \quad (3.4)$$

где  $x$  — длина отрезка, выделенного пунктирной линией. С другой стороны,

$$r_0^2 = (b + \lambda/2)^2 - (b + x)^2. \quad (3.5)$$

Из (3.4) и (3.5) следует, что  $2ax = b\lambda - 2bx + (\lambda/2)^2$ .

Как правило, в оптике нас интересует случай  $a, b \gg \lambda, x$ . Поэтому, пренебрегая слагаемыми, пропорциональными  $\lambda^2$  и  $x^2$ , получим:

$$x = \frac{b\lambda}{2(a + b)}$$

и  $r_0^2 \approx 2ax$ , откуда

$$r_0 = \sqrt{\frac{\lambda ab}{a + b}}. \quad (3.6)$$

Формула (3.6) дает радиус нулевой зоны Френеля. В этой формуле  $\lambda$  — длина световой волны,  $a$  и  $b$  — длины отрезков, показанных на рис. 3.7. Аналогичным образом находим внешний радиус  $n$ -й зоны Френеля:

$$r_n = \sqrt{(n + 1) \frac{\lambda ab}{a + b}}. \quad (3.7)$$

Несложно убедиться, что площади всех зон Френеля одинаковы.

### Расчет внеосевой интенсивности

Описанный выше метод зон Френеля позволяет рассчитать поле только на оси круглого отверстия. Для расчета интенсивности в любой точке изображения его необходимо модифицировать.

Отличие любой внеосевой точки изображения от осевой заключается в том, что появляются подзоны, открытые лишь частично (см. рис. 3.8). Их вклад в общее поле, таким образом, необходимо уменьшить. Полагая ширину подзоны малой, можно считать, что вклад в результирующее поле пропорционален длине границы подзоны, открытой исходным отверстием.

Рассчитаем открытую часть подзоны. Для этого воспользуемся теоремой косинусов:  $(d/2)^2 = r^2 + r_i^2 - 2rr_i \cos(\alpha)$ , где  $d/2$  — диаметр отверстия, на котором происходит дифракция,  $r$  — расстояние от оси до точки наблюдения,  $r_i$  — радиус  $i$ -й подзоны,  $2\alpha$  — открытая часть дуги границы. Отсюда

$$\alpha = \arccos \frac{r^2 + r_i^2 - d^2/4}{2rr_i}. \quad (3.8)$$

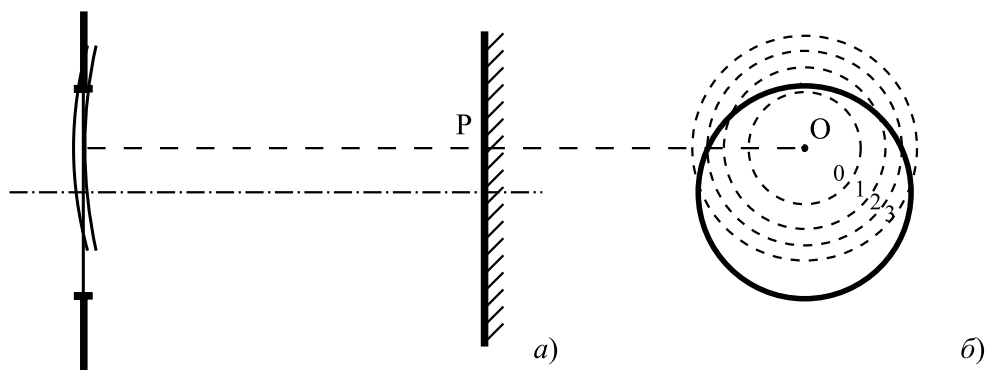


Рис. 3.8: К расчету внеосевой интенсивности при дифракции на круглом отверстии: а) расположение точки наблюдения P; б) вид отверстия и зон Френеля из точки наблюдения.

Для частично открытой подзоны ее вклад в результирующее поле необходимо домножить на отношение  $\alpha/\pi$ .

При  $r \leq d/2$  частично будут открыты те подзоны, размеры которых удовлетворяют следующему соотношению:

$$\frac{d}{2} - r \leq r_i \leq \frac{d}{2} + r. \quad (3.9)$$

При  $r \geq d/2$  это соотношение несколько меняется:

$$r - \frac{d}{2} \leq r_i \leq r + \frac{d}{2}. \quad (3.10)$$

Таким образом, несложная модификация метода зон Френеля позволяет получить интенсивность вне оси отверстия без расчетов дифракционного интеграла. Модифицированный метод был положен в основу модельной программы.

## 3.3 Основы теории дифракции Френеля

### 3.3.1 Интеграл Гюйгенса–Френеля

Рассмотрим источник света, который испускает монохроматическую световую волну. Световое поле в любой точке пространства также будет монохроматическим, его можно представить в виде [1, 3, 6]

$$E = \frac{1}{2} \mathcal{E} e^{i\omega t} + \text{к.с.},$$

где  $\omega$  — частота,  $\mathcal{E}$  — комплексная амплитуда колебаний. Согласно принципу Гюйгенса–Френеля, световое поле в точке P образуется в результате интерференции световых волн, испускаемых вторичными источниками–элементами поверхности  $\Sigma$ , то есть волнового фронта, создаваемого источником. Следовательно, можно записать:

$$\mathcal{E}(P) = \int_{\Sigma} \mathcal{E}(M) \frac{\exp(-ik\rho)}{\rho} K(\vartheta) d\sigma. \quad (3.11)$$

Здесь  $\mathcal{E}(P)$  и  $\mathcal{E}(M)$  — комплексные амплитуды поля в точке наблюдения P и в точке волнового фронта M соответственно,  $k = \omega/c = 2\pi/\lambda$  — волновое число,  $K(\vartheta)$  — «коэффициент наклона», учитывающий то, что вклад элемента  $d\sigma$  в результирующее поле зависит от ориентации данного элемента поверхности по отношению к направлению на точку наблюдения.

Интеграл (3.11) носит название интеграла Гюйгенса–Френеля. Формула построена на основе качественных физических соображений. Множитель  $e^{-ik\rho}/\rho$  в подинтегральном выражении описывает изменение амплитуды и фазы элементарной вторичной сферической световой волны. Наиболее существенен учет фазы, который и позволяет рассчитать результат интерференции вторичных волн. Функция  $K(\vartheta)$  в интеграле строго не определена. Френель полагал, что  $K(\vartheta)$  монотонно убывает от некоторого начального значения  $K(0)$  до нуля при изменении угла  $\vartheta$  от нуля до  $\pi/2$ . Для случая малых углов между нормалью к волновому фронту и направлением на точку наблюдения можно принять ее слабо изменяющейся и линейной.



### 3.4 Дифракция Фраунгофера

Эта программа позволяет моделировать и наблюдать дифракционную картину, получаемую в дальней зоне при освещении плоской монохроматической волной отверстий круглой и прямоугольной формы и щели, а так же график зависимости интенсивности дифракционной картины от радиальной координаты.

Интенсивность моделируемой картины рассчитывается следующим образом:

Рассмотрим дифракцию одномерной плоской волны на щели шириной  $d$ . Положим

$$\varepsilon_0(x) = \begin{cases} 1, & |x| \leq d/2 \\ 0, & |x| > d/2 \end{cases}$$

Интеграл Гюйгенса-Френеля запишется в виде:

$$\varepsilon_0(k_x) = \varepsilon_0 \int_{-d/2}^{d/2} \exp(ik_x x) dx = \varepsilon_0 d \operatorname{sinc} \left( \frac{k_x d}{2} \right)$$

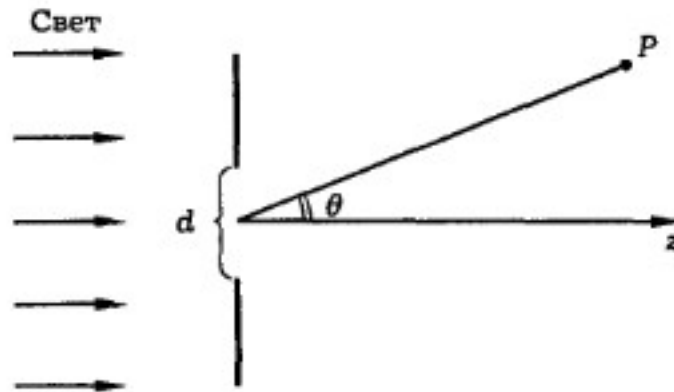


Рис. 3.9: К расчету картины дифракции света на щели

Найдем амплитуду поля:

$$\varepsilon(P) = \varepsilon_0 \frac{(i+1)d}{\sqrt{2\lambda b}} e^{-ikb} \operatorname{sinc}(k_x d/2)$$

и интенсивность поля в точке  $P$ :

$$I(P) = I_{max} \operatorname{sinc}^2(k_x d/2)$$

здесь  $I_{max} = \frac{I_0 d^2}{\lambda b}$ ,  $I_0 = \frac{c}{8\pi} |\varepsilon_0|^2$ . Тогда угловое распределение интенсивности в дальней зоне имеет вид:

$$I(\theta) = I_{max} \text{sinc}^2 \left( \frac{kd \sin \theta}{2} \right) = I_{max} \text{sinc}^2 \left( \frac{\pi d \sin \theta}{\lambda} \right)$$

Распределение интенсивности показано на рисунке 3.10

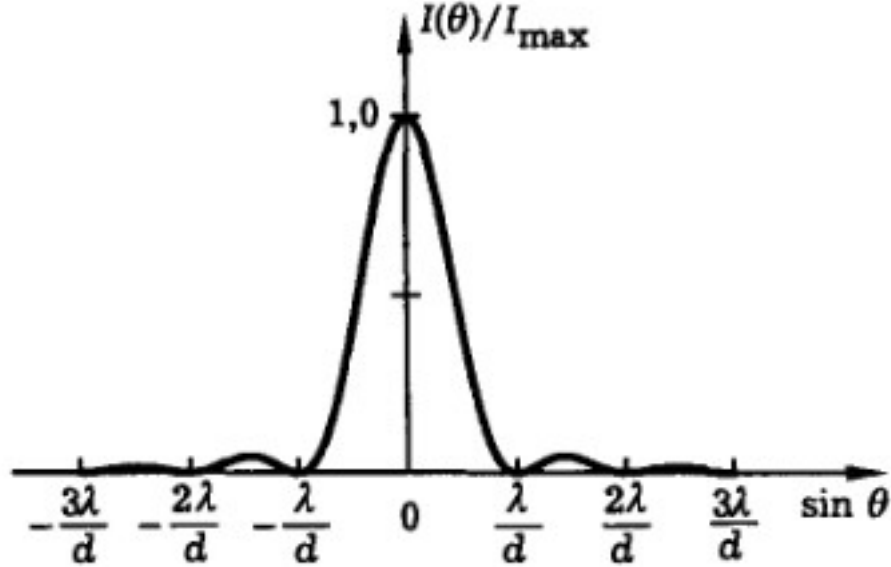


Рис. 3.10: Угловое распределение интенсивности дифракционной картины

Аналогично найдем распределение интенсивности дифракционной картины в дальней зоне для прямоугольного отверстия:

$$I(\theta, \psi) = I_{max} \text{sinc}^2 \left( \frac{\pi d_1 \sin \theta}{\lambda} \right) \text{sinc}^2 \left( \frac{\pi d_2 \sin \psi}{\lambda} \right)$$

где  $I_{max} = I_0 (d_1 d_2 / \lambda b)^2$ ,  $I_0$  – интенсивность падающей волны.

Для круглого отверстия:

$$I_\theta = I_{max} \left[ \frac{J_1(2\pi\theta R/\lambda)}{\pi\theta R/\lambda} \right]^2$$

где

$$I_{max} = I_0 \left( \frac{\pi R^2}{\lambda b} \right)^2$$

Интерфейс программы.

На рисунке приведено изображение окна программы.

В левом верхнем углу расположены вкладки для переключения между круглым отверстием, щелью и прямоугольным отверстием. В левом верхнем углу каждой вкладки приведена схема моделируемой установки. Под схемой расположены

элементы управления параметрами установки. Линейные параметры отверстия и длину волны падающего света изменяются в элементах редактирования. Кнопки “Ar” и “He-Ne” позволяют задать длину волны падающего света равной длине волны аргонового и гелий-неонового лазера. Ниже под кнопками расположено поле, показывающее цвет падающего излучения. Наблюдаемая дифракционная картина (наглядный результат моделирования) отображается в правой верхней части экрана. Под ней находится график зависимости интенсивности дифракционной картины от радиальной координаты. График имеет тот же самый пространственный масштаб, что и интерференционное изображение, что позволяет произвести количественное описание наблюдаемой картины.

## Заключение

1. На основе анализа психофизиологических, методических и эргономических требований к интерфейсам модельных программ разработана визуальная модель интерфейса, учитывающая возможности свободно распространяемой объектно-ориентированной кроссплатформенной библиотеки Qt.
2. Разработана и создана модельная программа на языке C++ с использованием библиотеки Qt и получены модельные изображения дифракции и интерференции некогерентного света.

# Литература

- [1] Ахманов С.А., Никитин С.Ю. Физическая оптика. — М., Изд-во МГУ, 1998.
- [2] Бланшет Ж., Саммерфилд М. Qt 4: Программирование GUI на C++. — М., КУДИЦ-ПРЕСС, 2008.
- [3] Борн М., Вольф Э. Основы оптики. — М., Наука, 1973.
- [4] Гажур И. А. Разработка компьютерных демонстраций по оптике с учетом методических и эргономических требований. Дипломная работа. — М., Физический факультет МГУ, 2006.
- [5] Данилова Н. Н. Психофизиология. — М., Аксепт-Пресс, 1998.
- [6] Ландсберг Г.С Оптика. — М., ФИЗМАТЛИТ, 2003.
- [7] Раскин Д. Интерфейс: новые направления в проектировании компьютерных систем. — СПб., Символ-Плюс, 2004.
- [8] Селиверстов А. В. Современные лекционные демонстрации по разделу “Волновая оптика” курса общей физики. Дисс. ... канд. пед. наук. — М., Физический факультет МГУ, 2005.
- [9] Ситаров В.А. Дидактика. — М., Издательский центр “Академия”, 2002.
- [10] Служба тематических толковых словарей. URL: <http://www.glossary.ru/>
- [11] Чурикова Ю.В. Разработка методических принципов построения демонстрационных программ и создание компьютерных модельных демонстраций к разделу “Оптика” курса общей физики. Дипломная работа. — М., Физический факультет МГУ, 2000.
- [12] Шлее М. Qt 4: Профессиональное программирование на C++. — СПб., БХВ-Петербург, 2007.