

Некоммерческое акционерное общество
АЛМАТИНСКИЙ УНИВЕРСИТЕТ ЭНЕРГЕТИКИ И СВЯЗИ

кафедра Компьютерных технологий

«Допущен к защите»
Заведующий кафедрой _____

(Ф.И.О., ученая степень, звание)

« _____ » _____ 20__ г.
(подпись)

ДИПЛОМНЫЙ ПРОЕКТ

На тему: Исследование свойств время-пространственной фокусировки дивергентной электромагнитной волны

Специальность _____

Выполнил (а) Мамариев Д.В. ВМ-10-3
(Фамилия и инициалы) группа

Научный руководитель _____
(Фамилия и инициалы, ученая степень, звание)

Консультанты:

по экономической части:

Брессева З.Д., сев. преподаватель
(Фамилия и инициалы, ученая степень, звание)
Брессева « 20 » 05 2014 г.
(подпись)

по безопасности жизнедеятельности:

Фридрих Н.С., д.т.н., профессор
(Фамилия и инициалы, ученая степень, звание)
Фридрих « 15 » 05 2014 г.
(подпись)

по применению вычислительной техники:

(Фамилия и инициалы, ученая степень, звание)
Сидель « 23 » _____ 20__ г.
(подпись)

(Фамилия и инициалы, ученая степень, звание)
« _____ » _____ 20__ г.
(подпись)

Нормоконтролер: Тусупов Д.М.
(Фамилия и инициалы, ученая степень, звание)
Тусупов « 02 » июль 2014 г.
(подпись)

Рецензент: _____
(Фамилия и инициалы, ученая степень, звание)
« _____ » _____ 20__ г.
(подпись)

Алматы 2014 г.

Некоммерческое акционерное общество
АЛМАТИНСКИЙ УНИВЕРСИТЕТ ЭНЕРГЕТИКИ И СВЯЗИ

Факультет информационных технологий
Специальность «Вычислительная техника и ПО»
Кафедра «Компьютерных технологий»

ЗАДАНИЕ

на выполнение дипломного проекта

Студент Мамодиев Дамис Валерьевич
(фамилия, имя, отчество)

Тема проекта Исследование свойств времяпротек-
ной фокусировки дугерной электродной
микро

утверждена приказом ректора № 115 от «24» сентября 20__ г.

Срок сдачи законченной работы «__» _____ 20__ г.

Исходные данные к проекту требуемые параметры результатов проектирования (исследования) и исходные данные объекта

Перечень подлежащих разработке дипломного проекта вопросов или краткое содержание дипломного проекта:

- создание программного продукта для исследования фокусирующих свойств дугерной электродной микро
- расчет осевого распределения электростатического потенциала и его производных до второго порядка
- определение времяпротекных характеристик
- отладка, тестирование и оптимизация ПП.
- построение таблиц и графиков зависимости кардинальных элементов и времяпротекных характеристик от распределения потенциала

Аннотация

Итогом данного дипломного проекта будет программный продукт, за основу которого взят расчет и исследование времяпролетных характеристик двумерной электронной линзы.

Главным объектом данного проекта будет электронная линза. Будет рассмотрено само понятие «электронной линзы», произведен расчет ее поля. Будут рассмотрены движения частиц в этих полях. Так же будут рассмотрены параксиальные свойства линзы и ее аберрации.

Annotation

The program product, which is based on calculations and research of time-of-flight characteristics of two-dimension electron lens, is going to be the result of this diploma work.

The main object of this project is electron two-dimension lens. Calculation of electron field of lens and researching of definition of electron lens will be performed. Movement of electrons in electron fields will be reviewed. Paraxial properties of lens and its aberrations will be reviewed.

Андапта

Осы дипломдық жұмыста электрондық оптика саласындағы «Двумерлық электрондық дүрбінің уақыттық ерекшеліктерін зерттеу» атты ПП ұсынған.

Электрондық оптика – біраз зерттелген физиканың бір саласы, сондықтан да оның программалық базасы да аз.

Дипломдық жұмыстың мақсаты – физиканың осы саласындағы программалық базасын жетімді және түсінікті программалармен көбейту.

Содержание

Введение.....	7
1. Аналитическая часть проекта	9
1.1 Описание предметной области	9
1.2 Обзор аналогов	9
1.3 Постановка задачи.....	11
1.4 Цель дипломного проекта	11
1.5 Цель и назначение программного обеспечения.....	11
1.6 Актуальность программного обеспечения	11
1.7 Выбор языка и средства программирования.....	11
1.8 Требования к программному обеспечению.....	12
2 Проектная часть.....	13
2.1 Понятие электронной линзы	13
2.2 Расчёт поля.....	16
2.2.1 Уравнение Лапласа	17
2.2.2 Общее представление ряда.....	17
2.3 Уравнения движения.....	18
2.4 Время пролета в двумерной электронной линзе.....	20
2.5 Электронно-оптические свойства.....	22
2.6 Используемые в работе численные методы	24
2.6.1 Метод Рунге-Кутты четвёртого порядка	25
2.6.2 Метод Симпсона для численного определения интегралов	25
2.7 Инструкция по применению	26
3. Техничко-экономическое обоснование проекта	44
3.1 Описание работы и обоснование необходимости	44
3.2 Трудовые ресурсы, используемые в работе	44
3.3 Расчет стоимости работы по проектированию и разработке	44
3.4 Расчет затрат на разработку программного обеспечения	50
3.5 Цена интеллектуального труда (программного продукта и НИР (ОКР)) ...	54
4. Технические решения безопасности труда.....	56
Заключение	68
Список использованной литературы.....	69
Приложение А	70

Введение

В настоящее время, в условиях очень быстрого развития техники ученые и просто исследователи должны наблюдать и правильно характеризовать явления, происходящие с материалами на микро-, субмикро- и атомном-уровнях. В этом им помогают простой визуальный осмотр, оптические приборы и электронные микроскопы с высокой разрешающей способностью, вплоть до 0,001мкм.

Настоящая революция в электронной оптике началась в двадцатых годах двадцатого века. Все началось с идеи использования потомков частиц. Именно не одного электрона, а целые пучки. В связи с этим данная область получила широкую основу для развития виденья и изучения сверхмалых объектов.

Для людей было нормой, что изображение объекта в науке есть не что иное, как поступление на глаз(или, если говорить о науке, на прибор) световых волн этого предмета(объекта).

С помощью потоков электрон, возможно получать более четкие, а самое главное увеличенные во много раз объекты.

Волновые свойства электронов не сильно отличаются от привычных световых или радио волн. Электроны ведут себя как настоящие частицы, обладают массой, траекторией движения, энергией и другими свойствами. На пример так ведут себя электроны в приборах типичного быта: телевизорах, электронных лампах, кинескопах и других бытовых устройств.

Сам принцип ускорения электрона основан на действии известной силы, создаваемой полем (электрическим) образованным между электродами. Причем ускорение прямо пропорциональное силе и обратно пропорциональное его массе.

При движении потоков электронов они эквивалентны электрическим токам, в связи, с чем взаимодействовать со сторонними магнитными полями. От сюда напрашивается вывод: электрические и магнитные поля, могут сильно влиять на скорость электронных частиц и их траектории. Поэтому именно эти поля управляют движением таких частиц(пучков частиц). На этой «почве» рождается область науки, называемая «электронная оптика» - изучающая траекторию движения электронных потоков, а так же расчетом элементов и устройств, которые создают необходимые для этого поля.

«Электронная оптика»- малоизученный раздел физики. Но с совершенствованием компьютерных технологий, расширяются возможности и методы изучения, а также их упрощения. В связи с этим программа считается одной из немногих новых программ в данной предметной области.

Объектом на протяжении всего дипломного проекта является электронная линза. И написание программы для подобного изучения существенно упрощает расчетную часть.

Программа рассчитывает времяпролетные характеристики двумерной электронной линзы.

Программа написана на объектно-ориентированном языке Delphi7. Простота и незамысловатость среды позволяет даже типичному пользователю ориентироваться в данной программе. Человек только вводит начальные данные, а программа в свою очередь выдает уже посчитанный результат. Затем еще и строит графики по полученным данным.

1. Аналитическая часть проекта

1.1 Описание предметной области

Электронная оптика - дисциплина, занимающаяся вопросами формирования, фокусировки и транспортировки пучков заряженных частиц, в частности электронов, в магнитных и электрических полях. Практическое применение-формирование пучка электронов, и управление им, например, в электронно-лучевых трубках.

Изображение отдельных атомов впервые было получено при помощи просвечивающего электронного микроскопа, в котором электроны с энергией несколько мега электрон-вольт, сформированные в пучки, дали не теоретическое, а наглядное изображение атомов.

Электронная оптика применяется для расчётов, проектирования и эксплуатации устройств, оперирующих пучками заряженных частиц, в первую очередь это ускорители, каналы транспортировки пучка и т.д.

Электронные линзы - устройства, создающие магнитные или электрические поля для фокусировки электронных пучков, их формирования и получения электронно-оптических изображений (аналогичные устройства для ионных пучков наз. ионными линзами). Их классифицируют по типу поля (магнитные, электростатические), по виду симметрии (осе-симметричные, цилиндрические, квадрупольные и др.) и по др. характерным признакам.

1.2 Обзор аналогов

Так как данная область физики малоизучена, соответственно программная часть ее плохо развита. Именно поэтому данную программу можно считать одной из новых.

Существуют аналогичные программы, такие как «OPAL».

«OPAL» - пакет программ по расчету (анализ качества, синтез из модулей, оптимизация, расчет технологических допусков, каталоги стекол широкого класса оптических изображающих систем - центрированных и децентрированных с переменными характеристиками и прочее) для IBM PC/DOS и ЕС ЭВМ/IBM 370. Разрабатывался по заказу компетентных органов в ЛИТМО на протяжении 70-х, 80-х и начала 90-х годов коллективом ученых и программистов под руководством профессора Родионова.

Программа имеет достаточно понятный и красочный интерфейс в DOSe. На рисунке 1.1 можно увидеть интерфейс программы, возможность настраивать длины волн, диафрагму, смещение и многое другое.

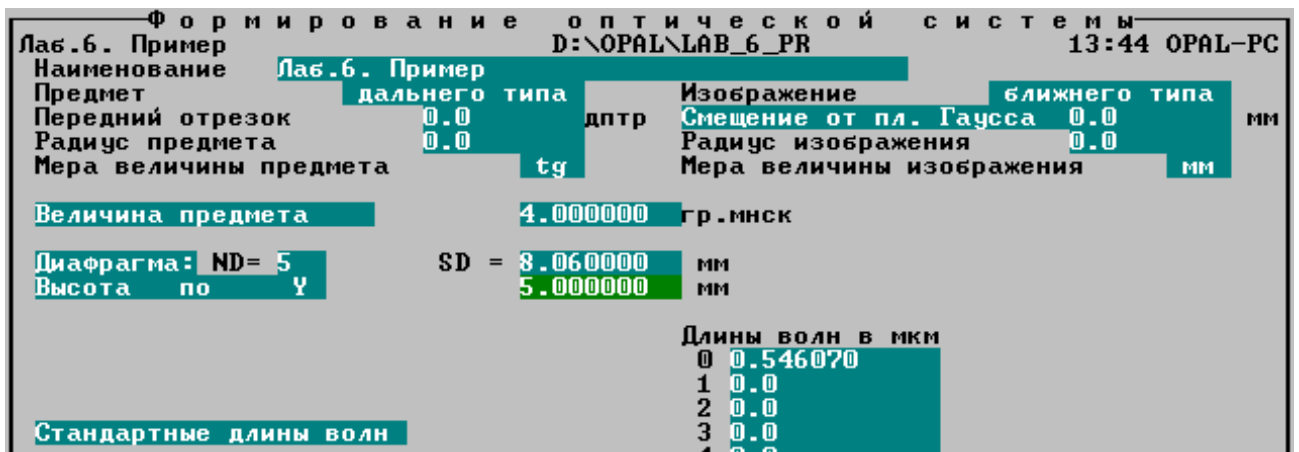


Рисунок 1.1 - Интерфейс программы OPAL

Так же есть возможность построения графического изображения движения частиц (Рисунок 1.2).

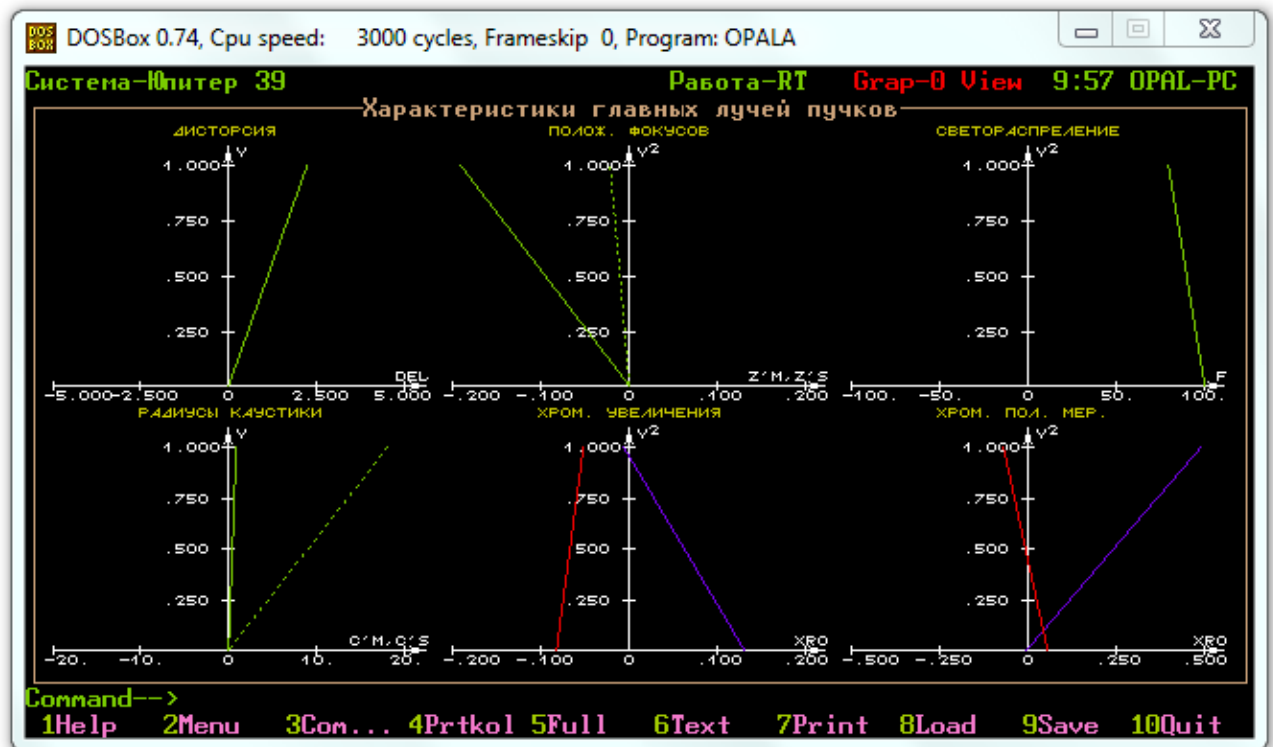


Рисунок 1.2 - Графики движения частиц электронной линзы в программе OPAL

Программа OPAL в отличие от данного дипломного проекта имеет сложности в установке. Системы DOS крайне устарели и на сегодняшний день, многим новым пользователям становится все труднее устанавливать и работать в данной системе. Именно поэтому данный программный продукт имеет актуальность, за счет крайне легкого и удобного интерфейса и более углубленного изучения конкретного вида линзы.

1.3 Постановка задачи

Главная задача программного продукта расчета характеристик времяпролетных характеристик двумерной линзы сводится к расчету поля линзы и поведения движения (траектории) пучка электронов в этом поле и самое главное – представления этих результатов этих решений графически.

Программа создается в несколько этапов:

- для расчета поля линзы создается небольшой графический интерфейс, который подкрепляется добавлением формул и начальных данных;
- далее создаются(вводятся) графики, которые строятся по данным расчета;
- далее компилируется код, ищутся ошибки, баги;
- закрепляется все графическим интерфейсом, понятным для любого пользователя.

1.4 Цель дипломного проекта

Целью дипломного проекта является разработка программного продукта по нахождению свойств времяпролетных характеристик двумерной электронной линзы понятного для типичного пользователя, обладающего хотя бы начальными знаниями в области электронной оптики.

1.5 Цель и назначение программного обеспечения

Целью программного обеспечения является предоставление возможности углубленного исследования времяпролетных характеристик электронного пучка при его прохождении через конкретную электронную линзу, в данном случае – двумерную.

Назначение программного обеспечения – снижение времени расчетов, связанных с нахождением характеристик поля линзы, времяпролетных характеристик, путем использования компьютера, а также предоставление графиков траектории электронного пучка.

1.6 Актуальность программного обеспечения

Данный дипломный проект считается актуальным, в связи с крайне малой степенью изученности данной области физики и неимением ее программной части, а также очень малого числа конкурентоспособных аналогов.

1.7 Выбор языка и средства программирования

Данной программное обеспечение написано на языке Delphi. Программа писалась в среде Borland Delphi 7.

Главными отличительными чертами данного языка Delphi являются императивность, структурированность и объектно-ориентация языка, т.е. созданные подпрограммы можно вызывать в любой части всей программы, что крайне облегчает программирование в данной среде.

Программ не требует вмешательства сторонних программ или другого программного обеспечения. Время установки занимает до 10 минут. Программу можно открывать в любом другом месте после ее создания.

1.8 Требования к программному обеспечению

Минимальные системные требования:

- операционная система: Microsoft Windows XP;
- процессор: 32-битный с тактовой частотой 700 МГц;
- оперативная память: 512 МВ RAM или выше;
- графическая карта VGA или выше;
- жёсткий диск: 20 МВ свободного места.

Рекомендуемые системные требования:

- операционная система: Microsoft Windows XP и выше;
- процессор: 32-битный с тактовой частотой 1000 МГц и выше;
- оперативная память: 1 GB RAM и выше;
- графическая карта VGA или выше;
- жёсткий диск: 20 МВ свободного места для установки и 100 МВ дополнительного свободного места.

2 Проектная часть

2.1 Понятие электронной линзы

Современное развитие многих отраслей естественных наук тесно связано с использованием электронно-вычислительных машин (ЭВМ), которые дают в руки исследователей эффективное средство для математического моделирования сложных задач науки и техники. В его основе лежит решение уравнений математической модели численными методами.

Одним из весьма распространенных применений численных методов являются расчеты электронно-оптических систем (ЭОС) - основных узлов электронно-лучевых приборов (ЭЛП). Повышение точности расчетов ЭОС способствует улучшению технических характеристик приборов. Это требует разработки эффективных алгоритмов для практических расчетов ЭЛП. Такие алгоритмы дают возможность заменить длительный и дорогостоящий технологический эксперимент расчетом на ЭВМ с выбором наилучшего решения разнообразных и сложных по геометрии ЭОС. Весьма актуальной задачей в настоящее время является автоматизация расчетов ЭОС на ЭВМ.

В основе расчета ЭОС лежит задача нахождения потенциала электростатического поля, которое создается системой электродов. С математической точки зрения она сводится к решению задачи Дирихле для уравнения Лапласа в пространстве.

Электронные линзы - устройства, создающие магнитные или электрические поля для фокусировки электронных пучков, их формирования и получения электронно-оптических изображений (аналогичные устройства для ионных пучков наз. ионными линзами). Их классифицируют по типу поля (магнитные, электростатические), по виду симметрии (осе-симметричные, цилиндрические, квадрупольные и др.) и по др. характерным признакам.

В электронных линзах воздействие на электронные пучки осуществляется электрическими или магнитными полями; эти линзы называются соответственно электростатическими или магнитными. Электронные линзы классифицируют по виду симметрии их поля и по его другим характерным признакам.

В некоторых электростатических электронных линзах одним из электродов служит катод, испускающий электроны (катодные линзы). Линза подобного типа ускоряет испущенные катодом электроны и формирует из них электронный пучок. Катодная электронная линза, состоящая лишь из двух электродов – катода и анода, не может фокусировать электронный пучок, именно с этой целью в систему вводят дополнительный электрод, который называется фокусирующим (Рисунок 2.1).

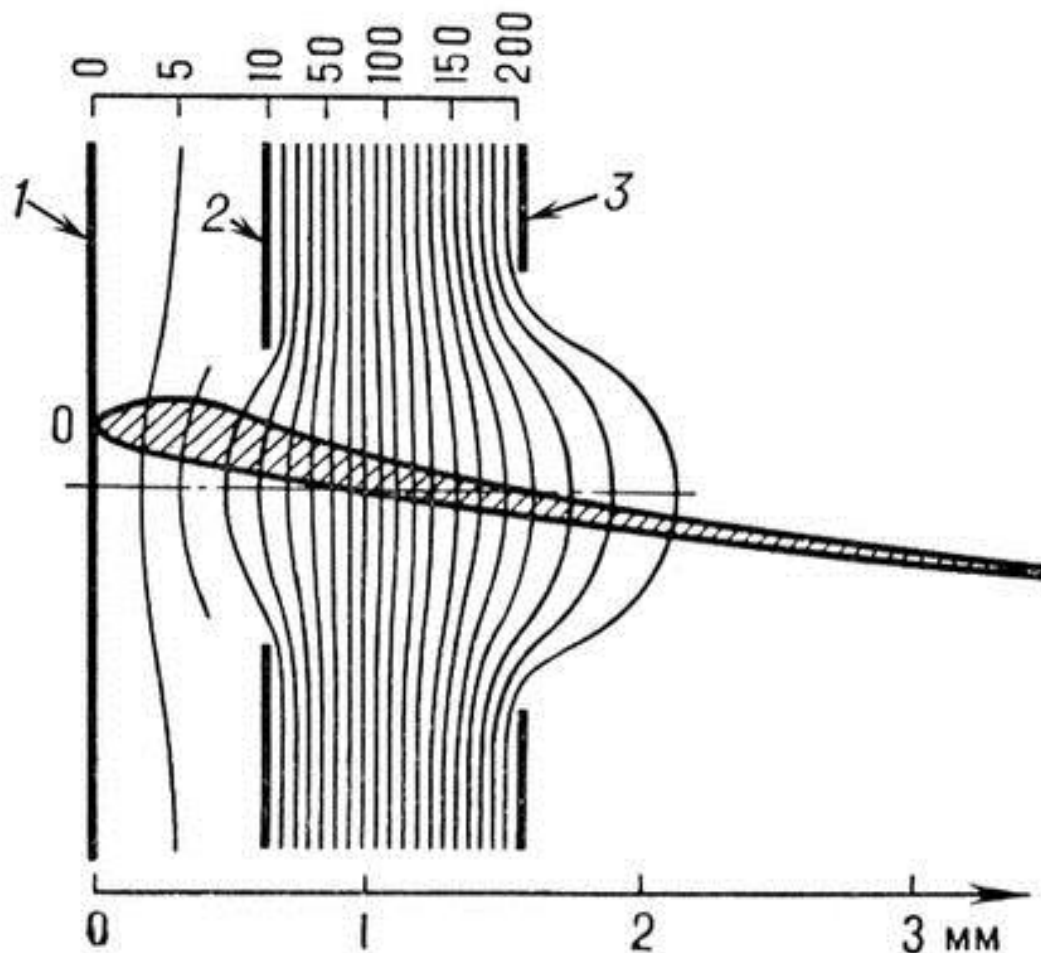


Рисунок 2.1 - Катодная электронная линза

где 1 - катод;

2 - фокусирующий электрод;

3 - анод.

В данном дипломном проекте, основным объектом расчета и изучения является двумерная электронная линза.

Двумерные (электронные) линзы являются одним из самых старых фокусирующим элементом в электронной оптике. Их название отталкивается от вида их поля: в области, где происходит исследование, далеко от краев оно практически не изменяется в одном из направлений, т.е. может считаться двумерным. В следствии чего фокусировка в данном направлении отсутствует, что делает линзы похожими (аналогичными) стеклянным цилиндрическим линзам. Отсюда и появилось название двумерных линз – цилиндрические.

Электроды двумерных цилиндрических линз представляют собой плоские пластины, один из размеров которых существенно больше другого. Они могут находиться параллельно оси пучка, совмещенной оси z (Рисунок 2.2) или перпендикулярно к ней (Рисунок 2.3).

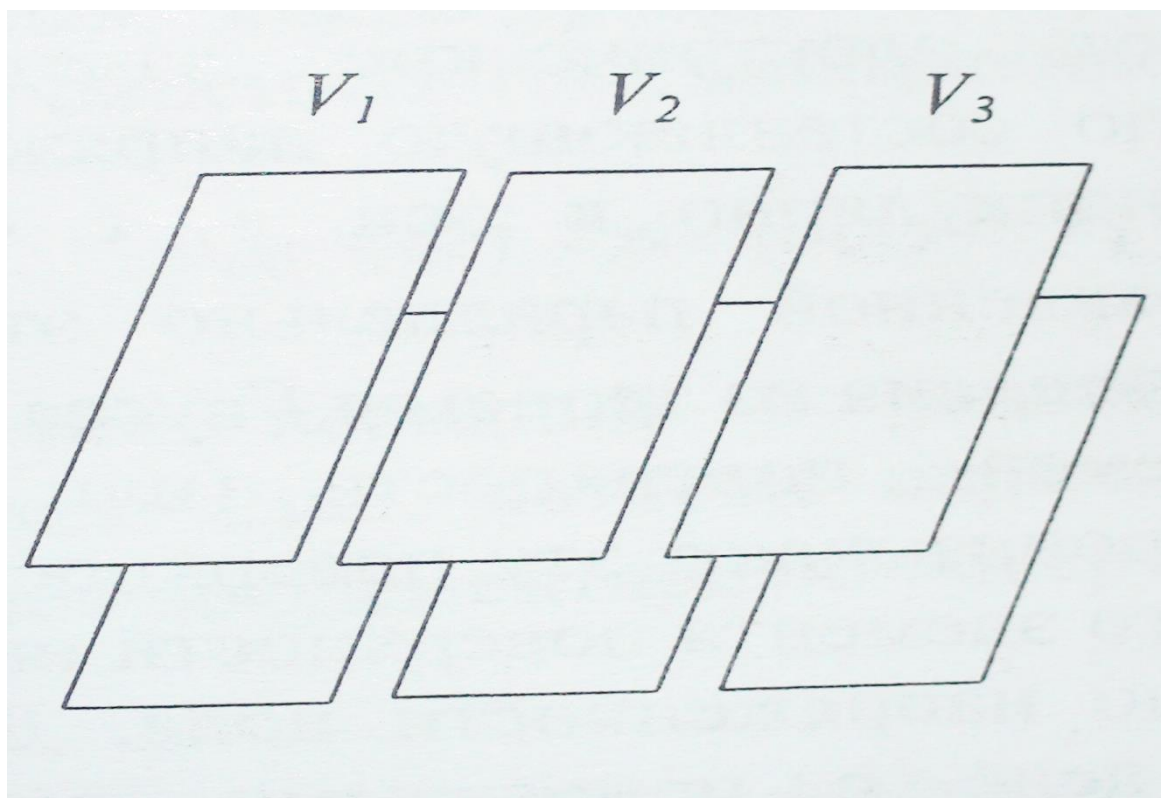


Рисунок 2.2 - Расположение электродов двумерной цилиндрической линзы, расположенные параллельно оси пучка.

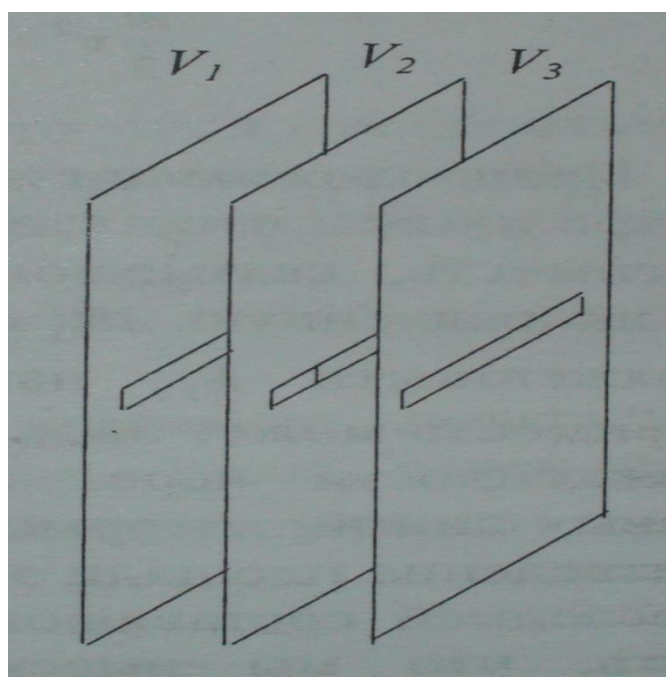


Рисунок 2.3 - Расположение электродов двумерной электрической линзы, расположенные перпендикулярно оси пучка.

На рисунке 2.2 каждый электрод линзы образуется парой пластин, симметрично расположенных относительно плоскости xz и

имеющих(находящихся) один потенциал. На рисунке 2.3 электродом является цельная пластина, в которой проделано щелевое отверстие. Из-за того, что размер отверстия в плоскости симметрии xz , называемой средней, гораздо больше его размера в перпендикулярном направлении, то такие двумерные линзы применяются для фокусировки ленточных пучков. Примерами таких служат ионные источники и призмные анализаторы.

Двумерная цилиндрическая линза, как и осесимметричная, может быть иммерсионной и одиночной. Различия состоят в количестве электродов. У иммерсионных линз минимальное число электродов - два. У одиночных - три.

Системы, состоящие из трехэлектродных двумерных линз, изображенные на рисунке 2.2, используются в призмных спектрометрах, а так же в виде отдельных элементов этих приборов, так и в составе электростатических призм.

Но при таких условиях они работают при телескопическом режиме.

Телескопический режим - режим, при котором падающий на линзу параллельный пучок остается параллельным, даже после прохождения поля. Поскольку все двумерные цилиндрические линзы всегда являются собирающими, телескопический режим может быть только при существовании промежуточного фокуса внутри системы. Подобные режим подбираются путем регулирования потенциала на электродах.

Если двумерная система является частью призмы и наряду с фокусировкой пучка обеспечивает его разделение по энергии, пучок заряженных частиц направляется в такую систему под значительным углом к плоскости xz .

2.2 Расчёт поля

Исследование и изучение любой линзы начинается с исследования и расчета ее поля. Но для начала рассмотрим луч в плоско – симметричном поле

$$m * \ddot{y} = -e * E_y \quad (2.1)$$

где \ddot{y} - ускорение;

E_y - составляющая напряженности электрического поля в направлении оси y .

Формулу 2.1 представим в дифференциальном виде

$$m * \ddot{y} = -q \frac{\partial \varphi}{\partial y}$$

Далее воспользуемся основным свойством плоско-симметричного поля

$$\varphi(y, z) = \varphi(-y, z)$$

2.2.1 Уравнение Лапласа

Уравнение Лапласа - дифференциальное уравнение в частных производных. В трёхмерном пространстве уравнение Лапласа запишется как сумма трёх частных производных по каждой оси

$$\Delta\varphi = \frac{\partial^2\varphi}{\partial x^2} + \frac{\partial^2\varphi}{\partial y^2} + \frac{\partial^2\varphi}{\partial z^2} = 0 \quad (2.1)$$

Так как в данном проекте используется двумерная линза, то в данном случае уравнение Лапласа будет выглядеть следующим образом

$$\Delta\varphi = \frac{\partial^2\varphi}{\partial x^2} + \frac{\partial^2\varphi}{\partial y^2} = 0$$

Объясняется это тем, что в одном из направлений линзы отсутствует фокусировка, в связи с совпадением движения.

2.2.2 Общее представление ряда

Потенциал φ не зависит от азимутального угла, то есть является функцией координат y и z

$$\frac{\partial}{\partial y} \left(y \frac{\partial\varphi}{\partial y} \right) + \frac{\partial}{\partial z} \left(y \frac{\partial\varphi}{\partial z} \right) = 0 \quad (2.2)$$

После деления на r и некоторого преобразования (2.2) можно записать в виде

$$\frac{4}{2y} \frac{\partial}{\partial y} \left(\frac{y^2}{2y} \frac{\partial\varphi}{\partial y} \right) + \frac{\partial^2\varphi}{\partial z^2} = 0 \quad (2.3)$$

Если положить $y^2 = u$, то

$$4 \frac{\partial}{\partial u} \left(u \frac{\partial\varphi}{\partial u} \right) + \frac{\partial^2\varphi}{\partial z^2} = 0 \quad (2.4)$$

Отсюда видно, что φ зависит от $u=y^2$ и от z . Поэтому можно написать

$$\varphi(y, z) = \Phi(z) + y^2\Phi_2(z) + y^4\Phi_4(z) + \dots \quad (2.5)$$

где $\Phi(z) = \varphi(0, z)$ обозначает осевое распределение потенциала, т.е. для $y=0$. Если ряд, представленный в формуле (2.5), подставить в уравнение Лапласа

(2.2), то, сравнивая коэффициенты при одинаковых степенях r , получим значения функций Φ_2, Φ_4, \dots так, что $\varphi(y, z)$ примет вид

$$\varphi(y, z) = \Phi(z) - \frac{y^2}{4}\Phi''(z) + \frac{y^4}{64}\Phi^{IV}(z) - \dots \quad (2.6)$$

Для целей электронной оптики достаточно знать первые три члена, записанные в (2.6).

Таким образом, потенциал в пространстве определяется его распределением на оси $\Phi(z) = \varphi(0, z)$.

Осевое распределение потенциала для системы, состоящей из двух соосных цилиндров равного диаметра d с бесконечно малым зазором между ними, может быть записано в виде

$$\Phi(z) = \frac{1}{2}[(V_1 + V_2) + (V_2 - V_1)U(z)] \quad (2.7)$$

Здесь V_1 и V_2 – потенциалы на первом и втором электродах, а $U(z)$ определяется формулой

$$U(z) = th\left(\frac{2,636z}{d}\right) \quad (2.8)$$

2.3 Уравнения движения

Уравнение движения записываются в виде второго закона Ньютона и закона сохранения энергии

$$m\ddot{y} = -eE_y \quad (2.9)$$

где \ddot{y} – ускорение;

E_y – составляющая напряженности электрического поля в направлении оси y .

Формулу 2.9 представим в дифференциальном виде

$$m\ddot{y} = -q \frac{\partial \varphi}{\partial y},$$

$$\frac{m}{2} \left[\left(\frac{dz}{dt} \right)^2 + \left(\frac{dr}{dt} \right)^2 \right] = -q(\varphi + \varepsilon) \quad (2.10)$$

где ε – величина, являющаяся разбросом энергии.

В разделе физики «механика», привычным для нас, является изучение движения отдельной единичной материальной точки. В отличие от механической физики, в оптической рассматривается пучок частиц, в следствие чего результатом будет траектория не одной частицы, а целого семейства.

Тогда, появляется необходимость перехода из уравнения движения путем расчетов перейти к уравнениям траекторий, путем исключения времени из формулы и перехода к зависимости от меридианной координаты z . Применим соотношения

$$\frac{d}{dt} = \frac{d}{dz} \frac{dz}{dt} = \dot{z} \frac{d}{dz},$$

$$\dot{y} = \frac{dy}{dt} = \dot{z} \frac{dy}{dz} = \dot{z} y',$$

$$\frac{m}{2} \dot{z}^2 (1 + y'^2) = -e\Phi(z),$$

$$\dot{z} = k \sqrt{\frac{\Phi(z)}{1 + y'^2}},$$

$$k = \sqrt{-\frac{2e}{m}}$$

Так как по условию параксиального приближения y и y' малы, то квадраты этих величин будут и подавно маленькими величинами. Отсюда следует, что

$$\dot{z} = k\sqrt{\Phi(z)},$$

$$\ddot{y} = \frac{d}{dt}(\dot{y}) = \dot{z} \frac{d}{dz}(\dot{z} y') = \dot{z} \left(\frac{d\dot{z}}{dz} y' + \dot{z} \frac{d}{dz} y' \right) = \frac{y'}{2} \frac{d}{dz}(\dot{z}^2) + \dot{z}^2 y'' \quad (2.11)$$

Тогда

$$\ddot{y} = \frac{y'}{2} \frac{d}{dz}(\dot{z}^2) + \dot{z}^2 y'',$$

$$\frac{d}{dz}(\dot{z}^2) = \frac{d}{dz}[k^2\Phi(z)] = k^2 \frac{d\Phi(z)}{dz},$$

$$\ddot{y} = k^2 \left[\frac{r'}{2} \frac{d\Phi(z)}{dz} + \Phi(z)y'' \right] \quad (2.12)$$

Из (2.10) известно, что

$$m\ddot{y} = -e \frac{\partial \varphi}{\partial y}$$

$\frac{\partial \varphi}{\partial y}$, в свою очередь, равен

$$\frac{\partial \varphi}{\partial y} = -\frac{y}{2} \Phi''(z) + \frac{r^3}{16} \Phi^{IV}(z) + \dots \quad (2.13)$$

Второй член будет настолько малым, что им и последующими членами ряда можно пренебречь. Значит

$$\ddot{y} = \frac{k^2}{2} \left[-\frac{y}{2} \Phi''(z) \right]$$

Отсюда следует, что

$$\frac{d^2 y}{dz^2} + \frac{\Phi'}{2\Phi} \frac{dy}{dz} + \frac{\Phi''}{4\Phi} y = 0 \quad (2.14)$$

Уравнение (2.14) можно записать и так

$$\Phi y'' + \frac{\Phi' y'}{2} + \frac{\Phi'' y}{4} = 0 \quad (2.15)$$

Таким образом, при удалении (отдалении) электрона от оси получилось линейное дифференциальное уравнение второго порядка. Данное уравнение впервые было получено Г. Бушем в 1926 году. Подобное уравнение (2.15) является линейным однородным дифференциальным уравнением второго порядка. Из теории известно, что общее решение данного уравнения есть суперпозиция двух его линейно независимых частных решений

$$y = c_1 s(z) + c_2 t(z) \quad (2.16)$$

где c_1 и c_2 – произвольные постоянные.

2.4 Время пролета в двумерной электронной линзе

Для вывода формулы расчета времени пролета частицы используем формулу закона сохранения энергии

$$\frac{mv^2}{2} = \frac{m}{2}(\dot{y}^2 + \dot{z}^2) = -q(\varphi + \varepsilon)$$

где m - масса частицы;

q - заряд частицы;

φ - потенциал электрического поля.

В данной системе координат потенциал φ не зависит от ψ и является функцией от y и z

$$\varphi = \varphi(y, z) = \Phi(z) - \frac{y^2}{4} \Phi''(z) + \frac{y^4}{64} \Phi^{(4)}(z)$$

В связи с тем, что $y=yz$ и ограничившись величинами не выше второго порядка

$$\dot{z}^2(1 + \dot{y}^2) = K(\varepsilon + \Phi(z) - \frac{y^2}{4} \Phi''(z))$$

где $K = \sqrt{-\frac{2q}{m}}$.

Откуда

$$\dot{z} = K \sqrt{\frac{\varepsilon + \Phi(z) - \frac{y^2}{4} \Phi''(z)}{(1 + \dot{y}^2)}}$$

$$\dot{z} = K \sqrt{\frac{\Phi(z) \left[1 + \frac{\varepsilon}{\Phi(z)} - \frac{y^2}{4} \Phi''(z) \right]}{(1 + \dot{y}^2)}}$$

Разложим результат по малым величинам y, y', ε получим $\dot{y} = y' \dot{z}$

$$\dot{z} = K \sqrt{\Phi(z)} \left[\varepsilon + \Phi(z) - \frac{y^2}{4} \Phi''(z) \right]^{\frac{1}{2}} (1 + \dot{y}^2)^{-\frac{1}{2}}$$

Получим

$$\dot{z} = K\sqrt{\Phi(z)} \left[1 + \frac{\varepsilon}{2\Phi(z)} - \frac{y^2}{8\Phi(z)} \Phi''(z) - \frac{1}{2} \dot{y}^2 \right]$$

Так как $\dot{z} = \frac{dz}{dt}$, время пролета t будет равно

$$t = \int_{z_0}^z \frac{dz}{\dot{z}} = \frac{1}{K} \int_{z_0}^z \frac{dz}{\sqrt{\Phi(z)} \left[1 + \frac{\varepsilon}{2\Phi(z)} - \frac{y^2}{8\Phi(z)} \Phi''(z) - \frac{1}{2} \dot{y}^2 \right]}$$

Преобразуя и интегрируя, получим

$$t = \frac{1}{K} \int_{z_0}^z \frac{1}{\sqrt{\Phi(z)}} \left[1 - \frac{\varepsilon}{2\Phi(z)} + \frac{y^2}{8\Phi(z)} \Phi''(z) + \frac{1}{2} \dot{y}^2 \right] dz \quad (2.17)$$

Представим (2.17) в виде суммы

$$t = \frac{1}{K} \left\{ \int_{z_0}^z \frac{dz}{\sqrt{\Phi(z)}} - \frac{\varepsilon}{2} \int_{z_0}^z \frac{dz}{\Phi(z)\sqrt{\Phi(z)}} + \frac{1}{2} \int_{z_0}^z \frac{dz}{\Phi(z)\sqrt{\Phi(z)}} \left[\Phi(z)y'^2 + \frac{1}{\Phi''(z)y^2} \right] dz \right\} \quad (2.18)$$

Для удобного анализа функции (2.18) перепишем в виде

$$t = \frac{1}{K} (J_0 + J_\varepsilon + J_y)$$

где $J_0 = \int_{z_0}^z \frac{1}{\sqrt{\Phi(z)}};$

$$J_\varepsilon = -\frac{\varepsilon}{2} \int_{z_0}^z \frac{dz}{\Phi(z)\sqrt{\Phi(z)}};$$

$$J_y = \left[\Phi(z)y'^2 + \frac{1}{\Phi''(z)y^2} \right] dz.$$

2.5 Электронно-оптические свойства

Параксиальное приближение в геометрической оптике - рассмотрение только параксиальных лучей, то есть лучей идущих под малыми углами к главной оптической оси. Параксиальное приближение применимо во многих оптических приборах и существенно упрощает теоретические расчеты.

Так как в нашем случае мы имеем дело с линзой, то рассмотрение параксиальности будет выглядеть как рассмотрение совокупности приосевых электронных траекторий, образующих двухпараметрическое семейство кривых. Совокупность подобных траекторий создает электронно-оптическое изображение.

Данное рассмотрение возможно при исследовании однородного линейного дифференциального уравнение второго порядка. Однородного

линейного дифференциального уравнение второго порядка имеет, как известно, два независимых друг от друга решения. Комбинация двух этих частных решений является общим решением.

Рассмотрим два частных интеграла $s(z)$ и $t(z)$ дифференциального уравнения (2.15), которые в точке $z = z_0$ следующим условиям

$$\begin{cases} s(z_0) = 1, t(z_0) = 0, \\ s'(z_0) = 0, t'(z_0) = 1 \end{cases} \quad (2.19)$$

Данные выражения не являются приосевыми траекториями, но любая приосевая траектория может быть получена из их линейной комбинации

$$y(z) = c_1 s(z) + c_2 t(z) \quad (2.20)$$

где c_1 и c_2 – произвольные (соответственно малые) постоянные.

Соотношением

$$y(z_0) = c_1 s(z_0) + c_2 t(z_0) = c_1 \quad (2.21)$$

Определена постоянная c_1 в соответствии с (2.19). Значение параметра c_2 определяется дифференцированием (2.20) и подстановкой $z=z_0$

$$y'(z_0) = c_1 s'(z_0) + c_2 t'(z_0) = c_2, \quad (2.22)$$

$$y(z) = y_0 s(z) + y'_0 t(z) \quad (2.23)$$

Параметр семейства $c_2 = y'(z_0)$, который определяет начальное направление каждого луча, исходящего из P_0 пучка, обозначим как γ . Итак, пучок лучей, исходящих из P_0 электронов, определяется согласно (2.20), (2.21) уравнением

$$y(z) = y(z_0)s(z) + \gamma t(z) \quad (2.24)$$

Рассмотрим произвольный кусочек перпендикулярной к оси плоскости $z = \bar{z}$. Тогда разным значениям $\gamma(\gamma', \gamma'', \dots)$ соответствуют разные точки пересечения P'_1, P''_1 , координаты которых определяются уравнением

$$y(\bar{z}) = y(z_0)s(\bar{z}) + \gamma t(\bar{z}) \quad (2.25)$$

Если выбрать плоскость экрана так, чтобы при $z = z_1$ функция $g(z)$ имела следующую нулевую точку, то из (2.24) следует

$$y(z_1) = y(z_0)s(z_1) \quad (2.26)$$

Получилось уравнение, которое не зависит от семейства γ . Это говорит о том, что все траектории, вышедшие из P_0 соединятся в точке изображения P_1 , имеющей координаты $z=z_1$ и $y_1=y_0s_1$. Плоскость $z=z_1$ называют плоскостью изображения или плоскостью, сопряжённой с плоскостью предмета $z=z_0$. Отношение размера изображения r_1 к размеру предмета r_0 постоянно для всех сопряжённых точек. Точке предмета, расположенной на оси $(0, z_0)$ сопряжена точка изображения $(0, z_1)$, расположенная также на оси. Итак, приосевые траектории создают стигматическое изображение предметной плоскости на плоскость изображения с поперечным линейным увеличением

$$M = p(z_1) \quad (2.27)$$

Если известны две независимые электронные траектории, то с помощью их линейной комбинации можно найти «фундаментальные решения» $p(z)$ и $g(z)$, удовлетворяющие начальным условиям (2.40). Решение $g(z)$ сопрягает известной нулевой точке предмета $z=z_0$ следующую нулевую точку изображения $z=z_1$. Решение $p(z)$ даёт нам для $z=z_1$ соответствующее увеличение M (2.27), а производная решения $g(z)$ – поперечное угловое увеличение

$$\Gamma = t'(z_1) \quad (2.28)$$

Кардинальными элементами электронной линзы называют координаты фокусов и фокусные расстояния. Они также могут быть определены, если известны два каких-либо частных решения $p(z)$ и $g(z)$ уравнения (2.31). С учётом начальных условий, имеем

$$z(F_0) = z_0 + \frac{y'(z_1)}{s'(z_1)}, \quad (2.29)$$

$$z(F) = z_1 - \frac{y(z_1)}{s'(z_1)}, \quad (2.30)$$

$$f_1 = \frac{1}{s'(z_1)}, \quad (2.31)$$

$$f_0 = \sqrt{\frac{\Phi_0}{\Phi}} \quad (2.32)$$

Здесь $z(F_0)$ и f_0 – координата фокуса и фокусное расстояние предметного пространства, а $z(F)$ и f – пространства изображений.

2.6 Используемые в работе численные методы

2.6.1 Метод Рунге-Кутты четвёртого порядка

Дифференциального уравнения второго порядка решается исходя из формулы

$$y'' = f(x, y, y') \quad (2.33)$$

при начальных условиях

$$\begin{cases} y(x_0) = y_0, \\ y'(x_0) = y'_0 \end{cases} \quad (2.34)$$

Используя метод Рунге-Кутты четвёртого порядка

$$y_{k+1} = y_k + y'_k \Delta x + \frac{1}{6}(k_1 + k_2 + k_3)\Delta x, \quad (2.35)$$

$$y'_{k+1} = y'_k + \frac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4) \quad (2.36)$$

где $k_1 = f(x_k, y_k, y'_k)\Delta x$;

$$k_2 = f\left(x_k + \frac{\Delta x}{2}, y_k + y'_k \frac{\Delta x}{2}, y'_k + \frac{k_1}{2}\right)\Delta x;$$

$$k_3 = f\left(x_k + \frac{\Delta x}{2}, y_k + y'_k \frac{\Delta x}{2} + \frac{k_1}{4}\Delta x, y'_k + \frac{k_2}{2}\right)\Delta x;$$

$$k_4 = f\left(x_k + \Delta x, y_k + y'_k \Delta x + \frac{k_2}{4}\Delta x, y'_k + k_3\right)\Delta x.$$

Причём

$$x_k = x_0 + k\Delta x,$$

$$y_k = y(x_k),$$

$$y'_k = y'(x_k),$$

$$f_k = f(x_k, y_k, y'_k),$$

$$(k = 0, 1, 2, \dots)$$

2.6.2 Метод Симпсона для численного определения интегралов

Формула Симпсона (также Ньютона-Симпсона) относится к приёмам численного интегрирования. Получила название в честь британского математика Томаса Симпсона (1710—1761).

Основной принцип заключается в приближении подинтегральной функции на отрезке $[a, b]$ интерполяционным многочленом второй степени $p_2(x)$, то есть приближение графика функции на отрезке параболой. Метод Симпсона имеет порядок погрешности 4 и алгебраический порядок точности 3.

Формулой Симпсона называется интеграл от интерполяционного многочлена второй степени на отрезке $[a, b]$

$$\int_a^b f(x)dx \approx \int_a^b p_2(x)dx = \frac{b-a}{6} \left(f(a) + 4f\left(\frac{a+b}{2}\right) + f(b) \right) \quad (2.37)$$

где $f(a)$, $f\left(\frac{a+b}{2}\right)$ и $f(b)$ – значения функции в соответствующих точках (на концах отрезка и в его середине).

2.7 Инструкция по применению

При старте работы программы нам предоставляется вид основного меню программы (Рисунок 2.4). На нем отображены 4 функциональные кнопки действия. Такие как теория (немного теории о данной области), расчет программы, справка и выход.



Рисунок 2.4 - Основное меню программы

При активации кнопки «Выход» произойдет остановка активного режима программы и она просто закроется.

Код первой формы:

```

unit Unit2;

interface

uses
    Windows, Messages, SysUtils, Variants, Classes, Graphics,
Controls, Forms,
    Dialogs, StdCtrls, jpeg, ExtCtrls;

type
    TForm2 = class(TForm)
        Image1: TImage;
        Label1: TLabel;
        Button1: TButton;
        Button2: TButton;
        Button3: TButton;
        Button4: TButton;
        procedure Button1Click(Sender: TObject);
        procedure Button2Click(Sender: TObject);
        procedure Button3Click(Sender: TObject);
        procedure Button4Click(Sender: TObject);
    private
        { Private declarations }
    public
        { Public declarations }
    end;

var
    Form2: TForm2;

implementation

uses Unit3, Unit4, Unit5;

{$R *.dfm}

procedure TForm2.Button1Click(Sender: TObject);
begin
    form2.Close;
end;

procedure TForm2.Button2Click(Sender: TObject);
begin
    form3.show;
    form2.Visible := false;
end;

procedure TForm2.Button3Click(Sender: TObject);
begin
    form4.show;
    form2.Visible := false;
end;

```

```

procedure TForm2.Button4Click(Sender: TObject);
begin
form5.show;
end;

end.

```

Далее представлено окно справки, содержащее информацию о создателе и контактные данные (Рисунок 2.5).



Рисунок 2.5 - Окно справки программы

При нажатии на кнопку «назад» произойдет переход к основному меню программы, текущее окно будет автоматически закрыто.

При активации кнопки «Теория», находящейся в окне «Главное меню», откроется окно, содержащее теоретические факты о двумерных линзах. Код программы, написанный на форму справки, представлен ниже

```

unit Unit5;
interface

uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics,
  Controls, Forms,
  Dialogs, jpeg, ExtCtrls, StdCtrls;

type
  TForm5 = class(TForm)

```

```

    Label1: TLabel;
    Image1: TImage;
    Label2: TLabel;
    Label3: TLabel;
    Label4: TLabel;
    Button1: TButton;
    procedure Button1Click(Sender: TObject);
private
    { Private declarations }
public
    { Public declarations }
end;

var
    Form5: TForm5;

implementation

uses Unit2;

{$R *.dfm}

procedure TForm5.Button1Click(Sender: TObject);
begin
    form5.close;
    form2.show;
end;

end.

```

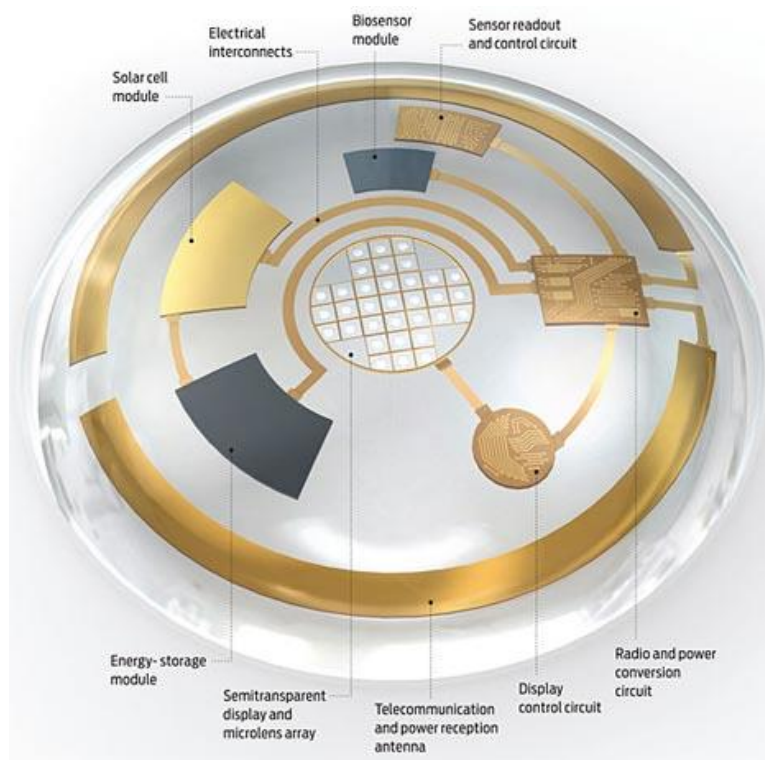


Рисунок 2.6 - Теоритические сведения

Из окна данной формы можно выйти как в главное меню, нажав кнопку «Выход в главное меню», так и из режима эксплуатации программы, нажав кнопку «Выход».

При нажатии на кнопку «Перейти к программе», находящейся в окне «Главное меню», пользователь будет переведён в форму, содержащую так называемую вступительную информацию. Данное окно представлено на рисунке 2.7.

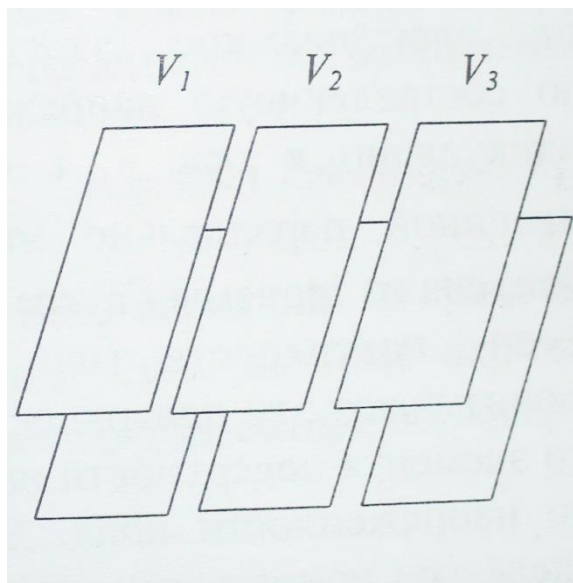


Рисунок 2.7 - Окно приветствия и указания действий

Из данного окна пользователь также может выйти из программы, вернуться в главное меню и перейти к следующей форме, которая содержит все вычислительные элементы. Эта форма называется «Окно расчётов» и представлена на рисунке 2.8.

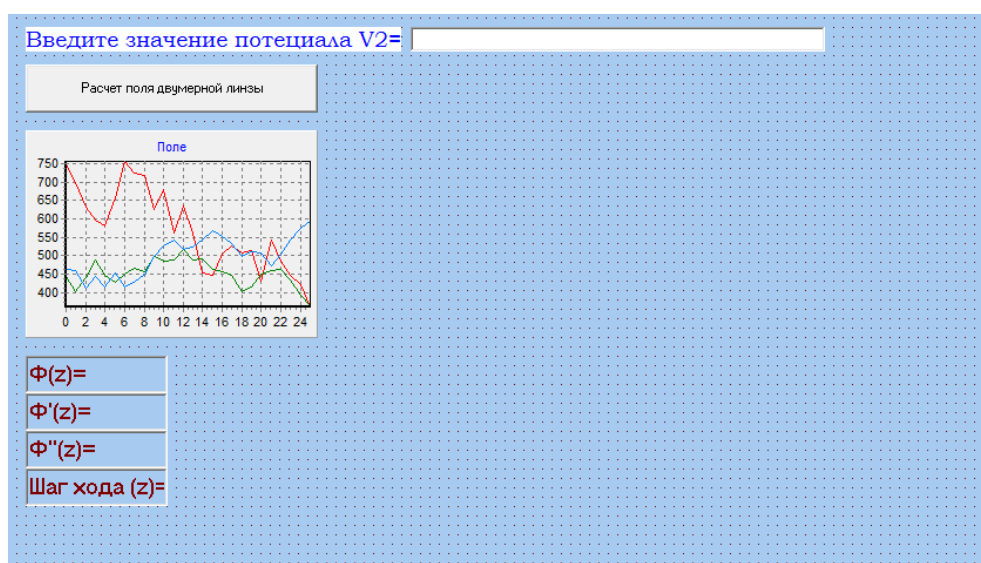


Рисунок 2.8 - Основное окно расчета программы

Код программы, написанный на форму вступительной информации, представлен ниже:

```
unit Unit3;

interface

uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics,
  Controls, Forms,
  Dialogs, StdCtrls;

type
  TForm3 = class(TForm)
    Label1: TLabel;
    Label2: TLabel;
    Button1: TButton;
    Button2: TButton;
    Button3: TButton;
    procedure Button2Click(Sender: TObject);
    procedure Button1Click(Sender: TObject);
    procedure Button3Click(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;
var
  Form3: TForm3;

implementation

uses Unit2, Unit1;

{$R *.dfm}

procedure TForm3.Button2Click(Sender: TObject);
begin
  form3.close;
  form2.close;
end;

procedure TForm3.Button1Click(Sender: TObject);
begin
  form2.Show;
  form3.Visible := false;
end;
```

```

procedure TForm3.Button3Click(Sender: TObject);
begin
form1.show;
form3.Visible := false;
end;

end.

```

Данная форма содержит все вычислительные элементы, такие как графики для расчёта поля, расчёта р- и g-решений, график построения траекторий, расчёт координат плоскости фокального изображения и гауссовой плоскости. Также присутствуют поля для вывода полученных значений, таких как коэффициенты, найденные по методу Рунге-Кутты, координаты фокальной плоскости и плоскости гауссова изображения.

При входе в данное окно программа запросит ввод значения потенциала V_2 .

Если значение V_2 не введено, кнопки «Рассчитать поле», «Расчёт р- и g-решений», а также кнопка «Расчёт траектории» будут неактивны.

Также на элементе формы edit1, содержащем в себе значение V_2 , установлена так называемая «защита от дурака».

«Защита от дурака» предусматривает невозможность заполнения поля значения V_2 какими-либо другими символами, кроме цифр. Например, если пользователь вводит в поле V_2 букву или какой-либо символ, кроме запятой или цифры, программа не позволит ему это сделать, а кнопки расчётов останутся неактивными. Или если пользователь ввёл какое-то значение, но потом передумал и стёр данные, кнопки расчётов также перейдут из активного состояния в неактивное.

Далее, при введённом значении V_2 , можно переходить к кнопкам расчётов.

При нажатии на кнопку «Рассчитать поле», программа произведёт расчёты, прописанные в отдельных процедурах. Такие же действия производятся и при нажатии на кнопки «Расчёт р- и g-решений» и «Расчёт траектории». Также при нажатии на соответствующие кнопки в отдельные списки выведутся результаты расчётов, такие как расчёт поля и коэффициентов по методу Рунге-Кутты.

Также на форме «Окно расчётов» представлена кнопка «Выход», при нажатии на которую программа выходит из режима эксплуатации.

На рисунке 2.7 представлен график расчёта поля при заданном значении V_2 .

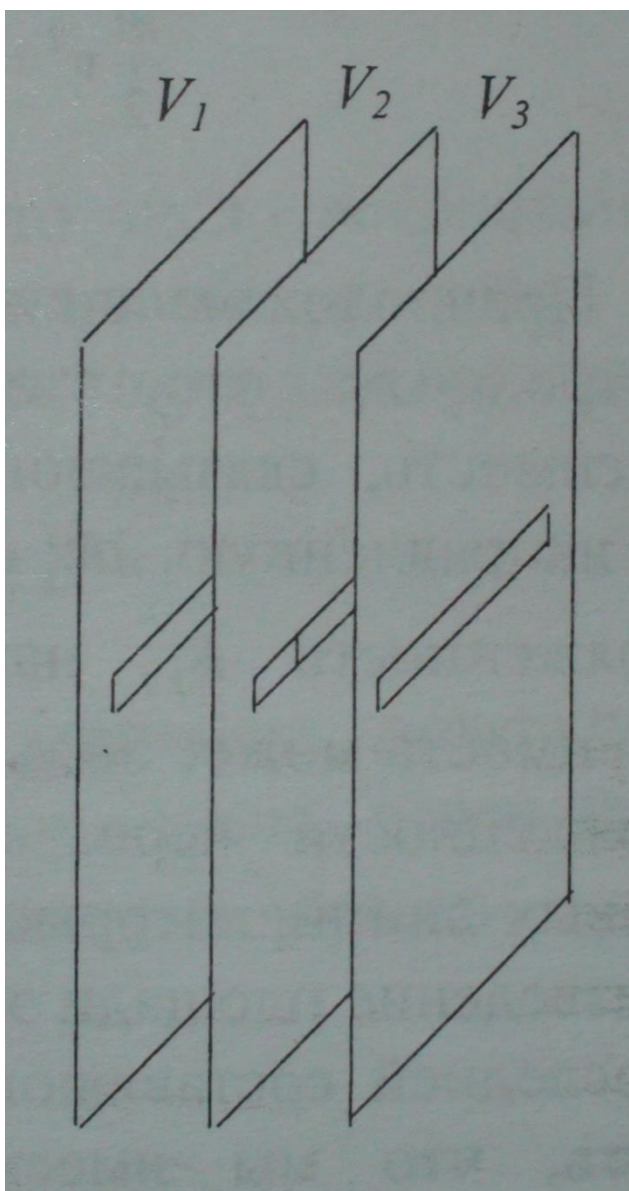


Рисунок 2.7 - Расчёт поля

На рисунке 2.8 представлен график расчета р- и г-решений.

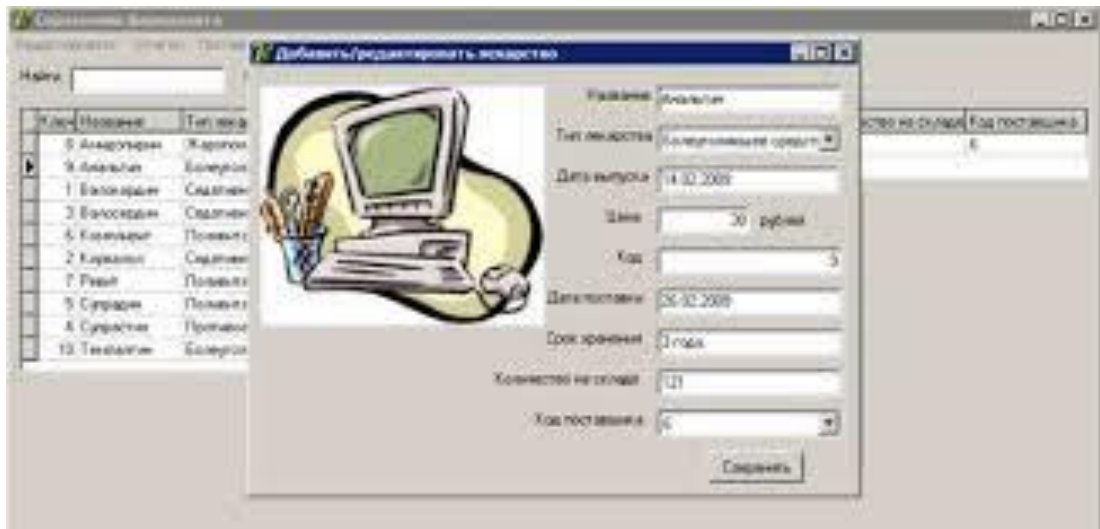


Рисунок 2.8 - Расчёт p- и g-решений

На рисунке 2.9 представлен график расчёта траектории, координаты фокальной плоскости и плоскости гауссова изображения.

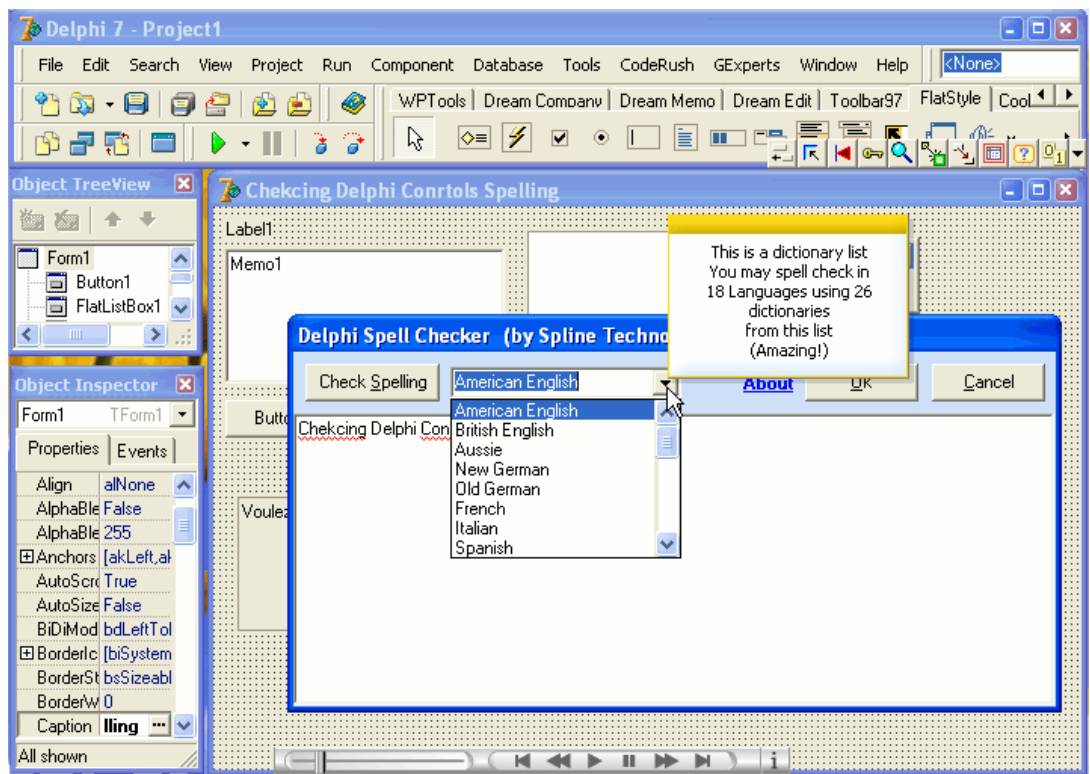


Рисунок 2.9 - Расчёт траектории и плоскостей изображений

Все графики и кнопки, представленные на форме «Окно расчётов», независимы друг от друга. Это значит, что пользователь может выбрать как построение только одного графика по заданному значению V_2 , так и два или все три графика.


```

Series2: TFastLineSeries;
Series3: TFastLineSeries;
Chart2: TChart;
Series4: TFastLineSeries;
Series5: TFastLineSeries;
Chart3: TChart;
Series6: TFastLineSeries;
Series7: TFastLineSeries;
Chart4: TChart;
Label6: TLabel;
Label1: TLabel;
Button2: TButton;
Button3: TButton;
Button4: TButton;
Series8: TFastLineSeries;
Series9: TFastLineSeries;
Series10: TFastLineSeries;
Label2: TLabel;
Label3: TLabel;
Label4: TLabel;
Label5: TLabel;
Memo1: TMemo;
Memo2: TMemo;
Memo3: TMemo;
Label7: TLabel;
Label8: TLabel;
Label9: TLabel;
Label10: TLabel;
Label11: TLabel;
Label12: TLabel;
Memo4: TMemo;
Memo5: TMemo;
procedure Button1Click(Sender: TObject);
procedure Button2Click(Sender: TObject);
procedure Button3Click(Sender: TObject);
procedure Button4Click(Sender: TObject);
procedure Edit1Change(Sender: TObject);
procedure Edit1KeyPress(Sender: TObject; var Key: Char);
private
  { Private declarations }
public
  { Public declarations }
end;
type Mass = array[0..2] of real;
procedure Pole(z, v2: Real; var FM: Mass);
procedure Polesh(z1, v2: Real; var FMSH: Mass);
procedure Runge(z, z1, v2: Real; var dm, em, gm, jm:
Mass);
procedure Trajectory(r0, r01: Real; dm, gm: Mass; var R:
Mass);
var
  Form1: TForm1;

```

```

z: Real;

implementation
uses Unit3, Unit2, Unit4;
{$R *.dfm}

procedure Pole(z, v2: Real; var FM: Mass);
var ez, ez1, uz, uz1, uz2, shz, chz, thz, thz1, thz2, fz,
fz1, fz2 :Real;
begin

ez := exp(2.618*z);
ez1 := 1/ez;
shz := (ez - ez1)/2;
chz := (ez + ez1)/2;
thz := shz/chz;
thz1 := 1/(chz*chz);
thz2 := (-2)*thz*thz1;

uz := thz;
uz1 := 2.618*thz1;
uz2 := 2.618*thz2;

fz := 0.5*((1 + v2) + (v2 - 1)*uz);
fz1 := 0.5*(v2 - 1)*uz1;
fz2 := 0.5*(v2 - 1)*uz2;
FM[0] := fz;
FM[1] := fz1;
FM[2] := fz2;
end;

procedure Polesh(z1, v2: Real; var fms: Mass);
var ezsh, ez1sh, uzsh, uz1sh, uz2sh, shzsh, chzsh,
thzsh, thz1sh, thz2sh, fzsh, fzsh1, fzsh2 :Real;
begin

ezsh := exp(2.618*z1);
ez1sh := 1/ezsh;
shzsh := 0.5*(ezsh - ez1sh);
chzsh := 0.5*(ezsh + ez1sh);
thzsh := shzsh/chzsh;
thz1sh := 1/(chzsh*chzsh);
thz2sh := (-2)*thzsh*thz1sh;

uzsh := thzsh;
uz1sh := 2.618*thz1sh;

```

```

uz2sh := 2.618*thz2sh;

fzsh := 0.5*((1 + v2) + (v2 - 1)*uzsh);
fzsh1 := 0.5*(v2 - 1)*uz1sh;
fzsh2 := 0.5*(v2 - 1)*uz2sh;
FMsh[0] := fzsh;
FMsh[1] := fzsh1;
FMsh[2] := fzsh2;

end;

Procedure Runge(z, z1, v2: Real; var dm, em, gm, jm: Mass);
var dm2, em2, em3, dm3, em4, dm4,
    gm2, jm2, gm3, jm3, gm4, jm4,
    g1, g2, g3, g4,
    k1, k2, k3, k4,
    h, h2, zmin, zmax: real;
    i, n :integer;
    FM, FMSH: Mass;
begin

h := 0.01;
h2 := h/2;

Pole(z, v2, FM);
Polesh(z1, v2, FMSH);

k1 := em[1]*h;
dm2 := dm[0] + em[0]*h2;
em2 := em[0] + k1/2;

k2 := ((-1)/(4*FMSH[0]))*(2*FMSH[1]*em2 + FMSH[2]*dm2)*h;
dm3 := dm[0] + em[0]*h2 + k1*h/4;
em3 := em[0] + k2/2;

k3 := ((-1)/(4*FMSH[0]))*(2*FMSH[1]*em3 + FMSH[2]*dm3)*h;
dm4 := dm[0] + em[0]*h + k2*h/4;
em4 := em[0] + k3;

k4 := ((-1)/(4*FM[0]))*(2*FM[1]*em4 + FM[2]*dm4)*h;

dm[0] := dm[0] + (em[0] + (k1 + k2 + k3)/6)*h;
em[0] := em[0] + (k1 + 2*k2 + 2*k3 + k4)/6;
em[1] := ((-1)/(4*FM[0]))*(2*FM[1]*em[0] + FM[2]*dm[0]);

g1 := jm[1]*h;
gm2 := gm[0] + jm[0]*h2;
jm2 := jm[0] + g1/2;

g2 := ((-1)/(4*FMSH[0]))*(2*FMSH[1]*jm2 + FMSH[2]*gm2)*h;

```

```

gm3 := gm[0] + jm[0]*h2 + g1*h/4;
jm3 := jm[0] + g2/2;

g3 := ((-1)/(4*FMSH[0]))*(2*FMSH[1]*jm3 + FMSH[2]*gm3)*h;
gm4 := gm[0] + jm[0]*h + g2*h/4;
jm4 := jm[0] + g3;

g4 := ((-1)/(4*FM[0]))*(2*FM[1]*jm4 + FM[2]*gm4)*h;

gm[0] := gm[0] + (jm[0] + (g1 + g2 + g3)/6)*h;
jm[0] := jm[0] + (g1 + 2*g2 + 2*g3 + g4)/6;
jm[1] := ((-1)/(4*FM[0]))*(2*FM[1]*jm[0] + FM[2]*gm[0]);

end;
procedure Trajectory (r0, r01: Real; dm, gm: Mass; var R:
Mass);
var z1, v2, sh: Real;
    jm, em : Mass;
begin
sh := 0.1;
Runge(z, z1, v2, dm, em, gm, jm);
R[0]:= r0*dm[0] + r01*gm[0];
end;

procedure TForm1.Button1Click(Sender: TObject);
var z, zmin, zmax, v2, h: real;
    i, n: integer;
    FM: Mass;
begin

v2 := strtofloat(edit1.Text);

zmin := -3;
zmax := 3;
h := 0.01;
z := zmin;

n := round((zmax - zmin)/h) + 1;

series1.clear;
series2.clear;
series3.clear;

for i := 1 to n do
begin

z := z + h;
Pole (z, v2, FM);

series1.AddXY(z, FM[0], '');
series2.AddXY(z, FM[1], '');

```

```

series3.AddXY(z, FM[2], '');

memo1.lines.Add(floattostr(FM[0]));
memo2.lines.Add(floattostr(FM[1]));
memo3.lines.Add(floattostr(FM[2]));

end;
end;

procedure TForm1.Button2Click(Sender: TObject);
begin
form1.Close;
form3.Close;
form2.Close;
form4.Close;
end;

procedure TForm1.Button3Click(Sender: TObject);
var zmin, zmax, v2, h, h2, z, z1,
    k1, k2, k3, k4,
    g1, g2, g3, g4,
    dm2, em2, em3, dm3, em4, dm4,
    gm2, jm2, gm3, jm3, gm4, jm4: real;
    i, n: integer;
    FM, dm, em, gm, jm, FMSH : Mass;
begin

v2 := strtofloat(edit1.Text);

zmin := -3;
zmax := 3;
h := 0.01;
h2 := h/2;
z := zmin;
z1 := zmin;

n := round((zmax - zmin)/h) + 1;

dm[0] := 1;
em[0] := 0;
dm[1] := em[0];
em[1] := 0;

gm[0] := 0;
jm[0] := 1;
gm[1] := jm[0];
jm[1] := 0;

series4.Clear;
series5.Clear;
series6.Clear;
series7.Clear;

```



```

for i := 1 to n do
begin

z := z + h;
z1 := z - h/2;

Pole (z, v2, FM);

Runge(z, z1, v2, dm, em, gm, jm);

series4.AddXY(z, dm[0], '');
series5.AddXY(z, em[0], '');

series6.AddXY(z, gm[0], '');
series7.AddXY(z, jm[0], '');

memo4.lines.add(floattostr(dm[0]));
memo5.Lines.add(floattostr(gm[0]));

end;
end;

procedure TForm1.Button4Click(Sender: TObject);
var zmin, zmax, rmin, rmax, rlmin, rlmax, r0, r01,
    v2, h, h2, sh, z, z1,
    k1, k2, k3, k4,
    g1, g2, g3, g4,
    dm2, em2, em3, dm3, em4, dm4,
    gm2, jm2, gm3, jm3, gm4, jm4,
    zf, zg: real;
    i, j, k, n, m, o: integer;
    FM, dm, em, gm, jm, r, FMSH : Mass;
begin

v2 := strtofloat(edit1.Text);

zmin := -3;
zmax := 3;
h := 0.01;
sh := 0.1;
h2 := h/2;
z := zmin;
z1 := zmin;
rmin := 0.2;
rmax := 1;
rlmin :=0.2;
rlmax := 1;
r0 := rmin;
r01 := rlmin;
n := round((zmax - zmin)/h) + 1;

```

```

o := round((rmax - rmin)/sh) + 1;

dm[0] := 1;
em[0] := 0;
dm[1] := em[0];
em[1] := 0;

gm[0] := 0;
jm[0] := 1;
gm[1] := jm[0];
jm[1] := 0;

series8.clear;

for i := 1 to n do
begin

z := z + h;
z1 := z - h/2;

r0 := r0 + sh;
r01 := r01 + sh;

Pole (z, v2, FM);

Runge(z, z1, v2, dm, em, gm, jm);

r[0] := r0*dm[0] + r01*gm[0];

series8.AddXY(z, r[0], '');

end;

zf := z - dm[0]/em[0];
zg := z - gm[0]/jm[0];

series9.clear;
series10.clear;

label4.caption := '';
label5.caption := '';

label4.caption := floattostrf(zf, ffGeneral, 4, 2);
label5.caption := floattostrf(zg, ffGeneral, 4, 2);

for i := 1 to 50 do
begin
series9.addX(zf, '');
series10.addX(zg, '');
end;
end;

```

```

    procedure TForm1.Edit1KeyPress(Sender: TObject; var Key:
Char);
    begin

    if not (key in ['0'..'9', ',', '-', #8])
    then key := #0;

    end;
    procedure TForm1.Edit1Change(Sender: TObject);
    begin

    if (Edit1.Text <> '') then Button1.Enabled := True;
    if (Edit1.Text <> '') then Button3.Enabled := True;
    if (Edit1.Text <> '') then Button4.Enabled := True;

    if (edit1.Text = '') then Button1.Enabled := False;
    if (edit1.Text = '') then Button3.Enabled := False;
    if (edit1.Text = '') then Button4.Enabled := False;

    end;

    end.

```

3. Технико-экономическое обоснование проекта

3.1 Описание работы и обоснование необходимости

Ключевым элементом любой области, будь то предприятие, бизнес и т.д. является программное обеспечение. ПО сильно упрощает различные вычисления и расчеты. А простой и приятный для восприятия интерфейс программного обеспечения помогает адаптироваться даже новичку.

Программное обеспечение исключает человеческий фактор ошибки. Программа не устает, не испытывает стресса и депрессии. Оно относительно надежно и имеет большую функциональность по сравнению с человеком. ПО служит для повышения как новых, так и уже существующих предприятий.

3.2 Трудовые ресурсы, используемые в работе

Над созданием проекта работают:

- руководитель - определяет задачи;
- инженер-разработчик - написание программы по расчету времяпролетных характеристик двумерной электронной линзы;
- консультанты по экономической части и по части ОБЖД сопровождение экономической части и основ правил безопасности жизнедеятельности;

Общее количество сотрудников и их заработная плата представлены в таблице 3.1.

Т а б л и ц а 3.1 - Количество задействованных в проекте работников, и их заработная плата

Исполнитель	Количество, человек	Месячная заработная плата, тенге
Руководитель	1	100000
Консультант по части “Экономика”	1	80000
Консультант по части “Безопасность жизнедеятельности”	1	80000
Инженер-разработчик	1	150000
Итого	4	410000

3.3 Расчет стоимости работы по проектированию и разработке

Создания программного продукта - тяжелый и долгий процесс, для его создания необходимы различные затраты: технические, интеллектуальные и

финансовые. Кроме того для такой зарплаты программиста, его опыт должен быть хотя бы 3-4 года. Затраты на разработку данного программного комплекса определяется по формуле

$$C = \text{ФОТ} + C_n + A + \text{Э} + C_{\text{пр}} + H \quad (3.1)$$

где ФОТ – фонд оплаты труда;

C_n – социальный налог;

A – амортизационные отчисления;

Э – затраты на электроэнергию;

$C_{\text{пр}}$ – прочие расходы;

H – накладные расходы.

Необходимый фонд оплаты труда рассчитывается по формуле и является суммой зарплат основной и дополнительной

$$\text{ФОТ} = Z_{\text{осн}} + Z_{\text{доп}} \quad (3.2)$$

где $Z_{\text{осн}}$ - основная заработная плата;

$Z_{\text{доп}}$ - дополнительная заработная плата.

Труд программиста - берется средняя заработная плата программиста и равна 150000 тенге.

Основная заработная плата рассчитывается по формуле и является суммой всех зарплат участвующих

$$Z_{\text{осн}} = Z_{\text{рук}} + Z_{\text{инж}} + Z_{\text{рук.э}} + Z_{\text{рук.бж}} , \quad (3.3)$$

$$Z_x = T_x * H_x \quad (3.4)$$

где T_x - суммарные затраты труда инженера, руководителя;

H_x - заработная плата инженера, руководителя за 1 час.

Основной показатель для составляющих затрат труда выражается формулой

$$Q = q * c \quad (3.5)$$

где $q=5200$, исходя из комплексного типа задачи;

c =стандартное значение описанное выше.

Результатом будет основной показатель исследования всего данного проекта. Значение коэффициента берется из таблицы 3.2.

Т а б л и ц а 3.2 - Значения коэффициента q

Тип задачи	Пределы изменений коэффициента
Задачи учета	от 1400 до 1500
Задачи оперативного управления	от 1500 до 1700
Задачи планирования	от 3000 до 3500
Многовариантные задачи	от 4500 до 5000
Комплексные задачи	от 5000 до 5500

Существуют 4 группы степени новизны программного продукта:

- группа А – разработка принципиально новых задач;
- группа Б – разработка оригинальных программ;
- группа В – разработка программ с использованием типовых решений;
- группа Г – разовая типовая задача.

Коэффициент “с” берем из таблицы 3.3, равный 1.3.

Т а б л и ц а 3.3 - Коэффициенты расчета трудоемкости

Язык программирования	Группа сложности	Степень новизны				Коэффициент B
		А	Б	В	Г	
Высокого уровня	1	1,38	1,26	1,15	0,69	1,2
	2	1,30	1,19	1,08	0,65	1,35
	3	1,20	1,10	1,00	0,60	1,5
Низкого уровня	1	1,58	1,45	1,32	0,79	1,2
	2	1,49	1,37	1,24	0,74	1,35
	3	1,38	1,26	1,15	0,69	1,5

Число команд Q можем определить из формулы (3.5)

$$Q = 5200 * 1,3 = 6760$$

После чего считаем примерное время создания программного продукта.

Для создания данного многомодульного программного обеспечения за основу берется техническое задание, которое выполняется по определенному расписанию, естественно с учетом сроков их выполнения. Часы берутся примерно от 3х до 5ти рабочих дней по 8 рабочих часов. Данное расписание-график с учетом дней, наименований и обозначений представлено в таблице 3.4. Так как люди работают по разному, то берется примерно от трех до пяти рабочих дней. После чего считаем примерное время создания программного продукта. Результатом будет являться примерно следующий график расписания работы.

Т а б л и ц а 3.4 – График выполненных работ по разработке проекта

Код работы	Наименование работы	Ожидаемая длительность, дни	Обозначение
1	Подготовка описания задачи	5	T _{ПЗ}
2	Описание задачи	5	T _{ОЗ}
3	Разработка алгоритма	5	T _А
4	Установка и запуск локального сервера	3	T _{ЛС}
5	Разработка модулей приложения	8	T _М
6	Разработка и создание интерфейса системы	5	T _{ИН}
7	Разработка основной части программы	10	T _{ОСН}
8	Тестирование базы данных и основных модулей с целью выявления ошибок	3	T _{ТЕСТ}
9	Составление технической документации	5	T _{ТЕХ}
10	Подготовка раздела “Экономика”	4	T _Э
11	Подготовка раздела “Безопасность жизнедеятельности”	3	T _{БЖ}

Различные участники работают в разное время. Поэтому производится расчет заработной платы каждого сотрудника, то есть сумму в день.

Формула для расчета заработной платы в день

$$D = Y / n \quad (3.6)$$

где Y – оклад работника в тенге;

n – количество дней в рабочем месяце (24 дня – шестидневная рабочая неделя).

Для руководителя

$$D = 70000 / 24 = 2916,67 \text{ тенге/день.}$$

Для консультанта по части “Экономика”

$$D = 45000 / 24 = 1875 \text{ тенге/день.}$$

Для консультанта по части “БЖД”

$$D = 45000 / 24 = 1875 \text{ тенге/день.}$$

Для инженера разработчика

$$D = 90000 / 24 = 3750 \text{ тенге/день.}$$

Заработная плата за один час рассчитывается по формуле

$$H = D / v \quad (3.7)$$

где D - заработная плата работника за один рабочий день;
 v - количество часов рабочего дня (8 часов).

Для руководителя

$$H = 2916,67 / 8 = 364,6 \text{ тенге/час.}$$

Для консультанта по части “Экономика”

$$H = 1875 / 8 = 234,4 \text{ тенге/час.}$$

Для консультанта по части “БЖД”

$$H = 1875 / 8 = 234,4 \text{ тенге/час.}$$

Для инженера разработчика

$$H = 3750 / 8 = 468,8 \text{ тенге/час.}$$

Время рассчитывается в человеко-часах, причем $T_{пз}$ берется по фактически отработанному времени, а время остальных этапов определяется расчетно, по условному числу команд Q .

Определяем время, затраченное на каждый этап создания программного продукта:

1. $T_{пз}$ (время на подготовку описания задачи), берется по факту и составляет (от 3-х до 5-ти дней по 8 часов).
2. $T_{оз}$ (время на описание задачи) определяется по формуле

$$T_{оз} = Q * B / (50 * K) \quad (3.8)$$

где В - коэффициент учета изменений задачи, коэффициент В, в зависимости от сложности задачи и числа изменений, выбирается в интервале от 1.2 до 1.5 (Таблица 3.5), равный 1.35;

К - коэффициент, учитывающий квалификацию программиста.

Выбираем значение коэффициента К из таблицы 3.5, равный 1.

Т а б л и ц а 3.5 - Коэффициенты квалификации программиста

Опыт работы	Коэффициент квалификации
До двух лет	0.8
2-3 года	1
3-5 лет	1.1 – 1.2
5-7 лет	1.3 – 1.4
более 7 лет	1.5 – 1.6

Применяя формулу 3.8 подсчитываем время на описание задачи

$$T_{\text{оз}} = 6000 * 1.4 / (50 * 1) = 168 \text{ чел-час.}$$

3. T_a (время на разработку алгоритма) рассчитываем по формуле

$$T_a = Q / (50 * K) \quad (3.9)$$

Тогда, применяя формулу 3.9, получаем

$$T_a = 6000 / (50 * 1) = 120 \text{ чел/час.}$$

4. $T_{\text{лс}}$ (время на установку и запуск локального сервера) определяется аналогично T_a

$$T_{\text{лс}} = 120 \text{ чел/час.}$$

5. $T_{\text{бс}}$ (время на разработку блок - схемы) определяется аналогично T_a по формуле 3.9 и составляет 120 чел-час.

6. T_m (время на разработку модулей приложения) определяется по формуле

$$T_m = Q * 1.5 / (50 * K)$$

Тогда

$$T_m = 6000 * 1.5 / (50 * 1) = 180 \text{ чел-час.}$$

7. $T_{\text{ин}}$ (время отладки и тестирования программы) определяется аналогично T_m по формуле 4 и составляет 180 чел-час.

8. T_d (время на оформление документации), берется по факту и составляет (от 3-х до 5-ти дней по 8 часов)

$$T_d = 24 \text{ чел / час.}$$

9. $T_{\text{осн}}$ (время на разработку основной части), определяется аналогично T_m по формуле 4 и составляет 180 чел-час.

10. $T_{\text{тест}}$ (время на тестирование), определяется аналогично T_m по формуле 4 и составляет 180 чел-час:

11. $T_{\text{тех}}$ (время на составление технической документации), берется по факту и составляет (от 3-х до 5-ти дней по 8 часов)

$$T_{\text{тех}} = 32 \text{ чел/час.}$$

12. $T_{\text{э}}$, $T_{\text{бж}}$ (время на подготовку разделов Экономика и БЖ), берется по факту и составляет (от 7-и до 10-ти дней по 8 часов)

$$T_{\text{э}} = T_{\text{бж}} = 56 \text{ чел/час.}$$

13. Т.к. в проекте задействованы руководители, заработная плата которых отличается, то необходимо рассчитать время $T_{\text{рук}}$. Время берется среднее и оно равно

$$T_p = 80 \text{ чел/час,}$$

$$T_{\text{ЭК}} = T_{\text{БЖД}} = 22 \text{ чел/час.}$$

Суммарные затраты труда рассчитываются как сумма составных затрат труда по формуле

$$T_{\text{ст}} = T_{\text{пв}} + T_{\text{оз}} + T_{\text{а}} + T_{\text{лс}} + T_{\text{бд}} + T_{\text{м}} + T_{\text{ин}} + T_{\text{осн}} + T_{\text{тест}} + T_{\text{тех}} + T_{\text{э}} + T_{\text{бж}}$$

$$T_{\text{ст}} = 40 + 168 + 120 + 120 + 120 + (180 * 4) + 24 + 32 + 56 + 56 = 1456 \text{ чел/час}$$

3.4 Расчет затрат на разработку программного обеспечения

Таким образом, суммарная основная заработная плата по формулам 3.3 и 3.4 составит

$$Z_{\text{осн}} = 1456 * (520,83 + 416,66 + 416,66 + 781,25) = 3\,109\,142,4 \text{ тг}$$

Дополнительная заработная плата в среднем определяется в размере 10% от основной заработной платы и рассчитывается по формуле

$$З_{\text{доп}} = З_{\text{осн}} * 10\% \quad (3.10)$$

$$З_{\text{доп}} = 3\,109\,142,4 * 0,1 = 310\,091,24 \text{ тг}$$

Общий фонд оплаты труда согласно формуле 4.2 составит

$$\text{ФОТ} = 3\,109\,142,4 + 310\,091,24 = 3\,420\,056,6 \text{ тг}$$

Социальный налог составляет 11% от ФОТ и рассчитывается по формуле

$$C_H = (\text{ФОТ} - \text{ПО}) * 11\% \quad (3.11)$$

где: ПО (пенсионные отчисления) составляют 10% от ФОТ и рассчитываются

$$\text{ПО} = \text{ФОТ} * 10\% \quad (3.12)$$

Размер пенсионных отчислений согласно формуле 3.12 составляет

$$\text{ПО} = 3\,420\,056,6 * 0,1 = 342\,005,66 \text{ тг}$$

Отчисления по социальному налогу согласно формуле 3.12

$$C_H = (2956878 - 295687,8) * 11\% = 338\,585,61 \text{ тг}$$

Амортизационные отчисления производятся по установленным нормам амортизации, выражаются в процентах к балансовой стоимости оборудования и рассчитываются по формуле

$$A = \frac{\text{Собор.} * N_A * N}{100 * 12 * t}$$

где N_A - норма амортизации;

Собор = 125 000 тг - первоначальная стоимость оборудования;

N - количество дней на выполнение работ;

t - общее время использования персонального компьютера.

Таблица 3.6 - Оборудование, необходимое разработки программного продукта

Наименование	Модель	Стоимость, тг
Процессор	2.0GHz core 2 duo Intel Core i3	80 000
Монитор	19-inch Samsung LED display; 2560-by-1440 resolution;16:9	30 000

Клавиатура	Es-9500	3000
Компьютерная мышь	Ms-6700	2000
Модем	TP-Link TL-WR940N	10 000
Итого		125 000

Норма амортизации (H_a), рассчитывается по формуле

$$H_a = ((C_{\text{обор.}} - C_{\text{ликв.}}) / (T_{\text{норм.}} * C_{\text{обор.}})) * 100\%$$

$$H_a = ((125\ 000 - 6566.5) / (4 * 125\ 000)) * 100 = 31.58\%$$

где $C_{\text{ликв.}} = C_{\text{обор.}} * 5.253\% = 0,05253 * 125000 = 6566.5$ тг - ликвидационная стоимость, составляет 5,253% от стоимости оборудования;

$T_{\text{норм.}}$ - нормативный срок службы (для персонального компьютера - 4 года).

Общее время использования персонального компьютера учитывает лишь время работы на компьютере и рассчитывается по формуле

$$t = T_a + T_{bc} + T_{осн} + T_{тест} \quad (3.13)$$

$$t = 120 + 120 + 180 + 180 = 600 \text{ ч}$$

Отсюда

$$A = (125\ 000 * 31.58 * 25) / (100 * 12 * 61) = 1348 \text{ тг}$$

Затраты на электроэнергию вычисляется по формуле

$$C_{\text{ээ}} = M * k_3 * T * C_{\text{кВт-ч}} \quad (3.14)$$

$$C_{\text{ээ}} = 0,6 * 0,8 * 600 * 14,935 = 432 \text{ тг}$$

где M - мощность ЭВМ (600 Вт=0,6 кВт);

k_3 - коэффициент загрузки (0.8);

$C_{\text{кВт.ч}} = 14,935$ тг – стоимость 1 кВт-час электроэнергии;

t - время работы, час.

Расходы на материалы и комплектующие, используемые в процессе написания программного продукта ($C_{\text{мик}}$), а также затраты на техническое обслуживание и ремонт ($C_{\text{то}}$) составляют, соответственно, 2,06% и 2,266% от стоимости оборудования - формулы

$$C_{\text{Мик}} = 0,0206 * C_{\text{обор}} \quad (3.15)$$

где $C_{\text{Мик}} = 0,0206 * 125000 = 2575$ тг.

$$C_{\text{ТО}} = 0,02266 * C_{\text{обор}} \quad (3.16)$$

где $C_{\text{ТО}} = 0,02266 * 125000 = 2832,5$ тг.

Накладные расходы, связанные с управлением и обслуживанием, содержанием и эксплуатацией оборудования и прочими дополнительными затратами на обеспечение процессов производства и обращения, составляют 72,1% от фонда оплаты труда, вычисляются по формуле

$$C_{\text{накл}} = 0,721 * \text{ФОТ} \quad (3.17)$$

$$C_{\text{накл}} = 0,721 * 3\,420\,056,6 = 2\,465\,860,8 \text{ тг}$$

Сводные результаты расчета себестоимости программного продукта предоставлены в таблице 3.7 и на рисунке 3.1, с указанием статьи расходов, суммы и их доли в общей стоимости разработки.

Т а б л и ц а 3.7 - Себестоимости программного продукта

Статья затрат	Сумма, тенге
Фонд оплаты труда	3 420 056.6
Затраты на электроэнергию	432
Социальный налог	338 585.61
Накладные расходы	2 465 860.8
Амортизационные отчисления	1348
Итого	6 226 283

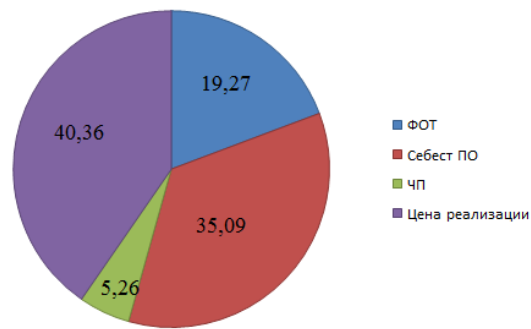


Рисунок 3.1 - Структурная диаграмма стоимости программного продукта по исследованию времяпролетных характеристик двумерной линзы.

3.5 Цена интеллектуального труда (программного продукта и НИР (ОКР))

Рассчитаем цену программного продукта программы

$$Ц_{пп} = З + П + НДС,$$

$$П = З * 0,15$$

$$ЧП = 6\,226\,283 * 15\% = 933\,955,9 \text{ тг}$$

Тогда, НДС от программного продукта составит

$$НДС = (З + П) * 12\%$$

$$НДС = 7160225,45 * 12\% = 859227,054 \text{ тг}$$

Тогда

$$Ц_{пп} = 6226283 + 933955,9 + 859227,054 = 8019465,954 \text{ тг}$$

Вывод

В данной части дипломной работы был произведен расчет себестоимости программного продукта, общий фонд оплаты труда, отчисления по социальному налогу, пенсионные отчисления, амортизационные отчисления,

также затраты на электроэнергию, на материалы и комплектующие, используемые в процессе написания программного продукта, накладные расходы. Большую часть занимает реализация данного проекта=40,36%. Далее идет себестоимость программного обеспечения. Далее идет фонд оплаты труда, что составляет 19,27% Все это составило цену реализации программного продукта и чистую прибыль от его реализации, равную 8019465,954 тг

4. Технические решения безопасности труда

4.1 Анализ потенциально опасных и вредных производственных факторов

80% пользователей ПК испытывают на себе физические расстройства, различной силы. Одни из таких расстройств: проблемы органов зрения и мышечные расстройства.

Для уменьшения данных воздействий на тело и организм человека специалистами был изучен и выведен ряд специальных требований к данной аппаратуре.

Производственный фактор, результатом которого в определенных условия является резкое ухудшение здоровья. Такой фактор, если в конечном итоге приводит к нетрудоспособности или заболеванию, называют вредным фактором. И вредные и опасные факторы делятся на следующие группы:

- физические;
- химические;
- психофизиологические;
- биологические.

В помещении лаборатории на программиста могут негативно действовать следующие физические факторы:

- повышенная и пониженная температура воздуха;
- чрезмерная запыленность и загазованность воздуха;
- повышенная и пониженная влажность воздуха;
- недостаточная освещенность рабочего места;
- превышающий допустимые нормы шум;
- повышенный уровень ионизирующего излучения;
- повышенный уровень электромагнитных полей;
- повышенный уровень статического электричества;
- опасность поражения электрическим током;
- блеклость экрана дисплея.

К химически опасным факторам, постоянно действующим на программиста, относят следующие: возникновение, в результате ионизации воздуха при работе компьютера, активных частиц.

Психологические вредные факторы, действующие на программиста в течении всего дня:

- нервно-эмоциональные перегрузки;
- умственное напряжение;
- перенапряжение зрительного анализатора.

4.2 Требования к помещениям для эксплуатации ПК

Помещения, в которых используются персональные компьютеры, обязаны соответствовать определенным правилам.

Такие помещения должны освещаться естественно и искусственно. Естественное осуществляется через окна (светопроемы), направленные на север и северо-восток. Естественное освещение должно осуществлять коэффициент естественной освещённости (КЕО) не ниже 1,2%.

Грубым нарушением будет считаться расположение рабочей зоны вблизи с помещениями с высоким уровнем шума и вибрации, которые имеют завышенные показатели нормируемые значения (заводы, цеха, мастерские).

Ограждающие объекты должны отвечать гигиеническим требованиям и обеспечивать нормальную звукоизоляцию шума (уровень шума < 50 дБа). Такого можно добиться с помощью звукопоглощающих материалов.

Помещения с персональными компьютерами должны быть оборудованы батареями (системами отопления) и нормальным воздухообменом.

Полимерные материалы, используемые для внутренней отделки интерьера помещений с ВДТ и ПЭВМ, должны быть разрешены для применения органами и учреждениями Государственного санитарно-эпидемиологического надзора. Запрещается для отделки внутреннего интерьера помещений с ВДТ и ПЭВМ применять полимерные материалы (древесностружечные плиты, слоистый бумажный пластик, синтетические ковровые покрытия др.), выделяющие в воздух вредные химические вещества.

Пол помещения обязан быть без ям и выбоин, ровным, максимально удобным для влажной уборки.

Помещения должны оборудоваться защитным заземлением.

4.3 Микроклимат рабочей зоны программиста

В помещениях с эксплуатацией ПК температура, относительная влажность и скорость движения воздуха на рабочих местах должны соответствовать действующим нормам микроклимата (Таблица 4.1).

Т а б л и ц а 4.1 - Оптимальные нормы микроклимата помещений с ПК

Период года	Категория работ	Температура воздуха, °С	Относительная влажность воздуха, %	Скорость движения воздуха, м/с
Холодный	Легкая - 1а	22-24	40-60	0,1
	Легкая - 1б	21-23	40-60	0,1
Теплый	Легкая - 1а	23-25	40-60	0,1
	Легкая - 1б	22-24	40-60	0,2

К категории 1а относятся работы, производимые сидя и не требующие физического напряжения, при которых расход энергии составляет до 120

кДж/ч; к категории 1б относятся работы, производимые сидя, стоя или связанные с ходьбой и сопровождающиеся некоторым физическим напряжением, при которых расход энергии составляет от 120 до 150 кДж/ч.

Нахождение вредных веществ в воздухе рабочего помещения, не должно превышать определенные нормы.

Микроклимат производственных помещений - это климат внутренней среды этих помещений, который определяется действующими на организм человека сочетаниями температуры, влажности и скорости движения воздуха. Рабочей зоной считается пространство высотой до 2 м над уровнем пола или площадки, на которой расположены рабочие места. Постоянным рабочим местом считается такое, где работающий находится более половины своего рабочего времени или более 2 ч непрерывно. Если же люди работают в различных местах рабочей зоны, то она вся считается рабочим местом.

4.4 Воздействие шума на программиста. Защита от шума

В помещениях с низким уровнем общего шума, каким является лаборатория, где работает программист, источниками шумовых помех могут стать вентиляционные установки, кондиционеры или периферийное оборудование для ЭВМ (плоттеры, принтеры и др.). Длительное воздействие этих шумов отрицательно сказывается на эмоциональном состоянии персонала.

Для того, чтобы добиться нормального уровня шума рекомендуется применять звукопоглощающее покрытие стен.

В качестве мер по снижению шума можно предложить следующее:

- облицовка потолка и стен звукопоглощающим материалом (снижает шум на 6 - 8 дБ);
- экранирование рабочего места (постановкой перегородок, диафрагм);
- установка в компьютерных помещениях оборудования, производящего минимальный шум;
- рациональная планировка помещения.

4.5 Опасность повышенного уровня напряженности электромагнитного поля

Электромагнитные поля, характеризующиеся напряженностями электрических и магнитных полей, наиболее вредны для организма человека. Основным источником этих проблем, связанных с охраной здоровья людей, использующих в своей работе автоматизированные информационные системы на основе персональных компьютеров, являются дисплеи (мониторы). Они представляют собой источники наиболее вредных излучений, неблагоприятно влияющих на здоровье программиста.

ЭВМ являются источниками таких излучений:

- ультрафиолетового 200 - 400 нм;

- видимого 400 - 700 нм,
- ближнего инфракрасного 700 - 1050 нм;
- радиочастотного 3 кГц - 30 МГц;
- электростатических полей.

Ультрафиолетовое излучение полезно в небольших количествах, но в больших дозах приводит к дерматиту кожи, головной боли, рези в глазах. Инфракрасное излучение приводит к перегреву тканей человека (особенно хрусталика глаза), повышению температуры тела. Уровни напряженности электростатических полей должны составлять не более 20 кВ/м. Поверхностный электростатический потенциал не должен превышать 500 В. При повышенном уровне напряженности полей следует сократить время работы за компьютером, делать пятнадцатиминутные перерывы в течение полутора часов работы и, конечно же, применять защитные экраны. Защитный экран, изготовляемый из мелкой сетки или стекла, собирает на себе электростатический заряд. Для снятия заряда экран монитора заземляют.

Может возникнуть опасность по уровням напряженности электромагнитного поля. Предельно допустимые значения характеристик ЭМП указаны в таблице 4.2.

Т а б л и ц а 4.2 - Предельно допустимые значения характеристик ЭМП

Наименование параметров	Допустимое Значение
Напряженность электромагнитного поля по электрической составляющей на расстоянии 50 см от поверхности видеомонитора	10 В/м
Напряженность электромагнитного поля по магнитной составляющей на расстоянии 50 см от поверхности видеомонитора	0,3 А/м
Напряженность электростатического поля не должно превышать: для взрослых пользователей	20 кВ/м
Напряженность электромагнитного поля на расстоянии 50 см вокруг ВДТ по электрической составляющей должна быть не более:	15 кВ/м
В диапазоне частот 5 Гц – 2 кГц;	25 В/м
В диапазоне частот 2 – 400 кГц	2,5 В/м
Плотность магнитного потока должна быть не более:	3,7 В/м
В диапазоне частот 5 Гц – 2 кГц;	250 нТл
В диапазоне частот 2 – 400 кГц	25 нТл
Поверхностный электростатический потенциал	500 В

Для предупреждения внедрения опасной техники все дисплеи должны проходить испытания на соответствие требованиям безопасности.

4.6 Требования к освещению

Работа за ПК относится к зрительным работам высокой точности для любого типа помещений.

Искусственное освещение в помещениях эксплуатации ПК должно соответствовать действующим нормам освещения (Таблица 4.3).

Т а б л и ц а 4.3 Нормирование освещенности лабораторий

1	2	3	4	5	6	7	8	9	10	11
Высокой Точности	От 0,3	Б	1	Более 70	300	100	40	15	3,0	1,0
	До 0,5		2	Менее 70	200	75	60	20	2,5	0,7

Освещенность на поверхности стола в зоне размещения рабочего документа должна быть 300-500 лк. Местное освещение не должно создавать бликов на поверхности экрана и увеличивать освещенность экрана более 300 лк.

Следует ограничивать прямую блескость от источников освещения, при этом яркость светящихся поверхностей (окна, светильники и т.п.), находящихся в поле зрения, должна быть не более 200 кд/м².

Следует ограничивать отраженную блескость на рабочих поверхностях (экран, стол, клавиатура и др.) за счет правильного выбора типа светильников и расположения рабочих мест по отношению к источникам естественного и искусственного освещения, при этом яркость потолка, при применении системы отраженного освещения, не должна превышать 200 кд/м².

В качестве источников света при искусственном освещении должны применяться преимущественно люминесцентные лампы типа ЛБ. Допускается применение ламп накаливания в источниках местного освещения.

Общее освещение следует выполнять в виде сплошных или прерывистых линий светильников, расположенных сбоку от рабочих мест, параллельно линии зрения пользователя при рядном расположении ПК. При периметральном расположении компьютеров линии светильников должны находиться ближе к переднему краю, обращенному к пользователю. Применение светильников без рассеивателей и экранирующих решеток не допускается.

Коэффициент запаса для осветительных установок общего освещения должен приниматься равным 1,4. Коэффициент пульсации не должен превышать 5 %.

4.7 Требования к ПК

Корпус ПК, клавиатура и другие блоки и устройства ПК должны иметь матовую поверхность одного цвета с коэффициентом отражения 0,4-0,6 и не иметь блестящих деталей, способных создавать блики.

Положение монитора должно быть таким, чтобы свет на него падал под углом. Экран монитора должен располагаться примерно на расстоянии 28-60 см от оператора, причем верхний край экрана должен находиться на уровне глаз. Рекомендуется по возможности уменьшать интенсивность света люминесцентных источников. Для монитора должны быть предусмотрены ручки регулировки яркости и контраста изображения, обеспечивающие возможность регулировки этих параметров от минимальных до максимальных значений.

4.8 Требования к организации режима работы с ПК

Режимы труда и отдыха при работе с ПК зависят от вида и категории трудовой деятельности (Таблица 4.4).

Экран монитора должен располагаться примерно на расстоянии 28-60 см от оператора, причем верхний край экрана должен находиться на уровне глаз.

Категории трудовой деятельности представлены в таблице 4.4.

Т а б л и ц а 4.4 - Время регламентированных перерывов в зависимости от продолжительности рабочей смены и вида трудовой деятельности

Категория работы с ПК	Уровень нагрузки за рабочую смену при видах работ	Суммарное время перерывов, мин			
		Считывание информации число знаков	Ввод информации число знаков	Работа в режиме диалога, ч	8-ми часовая смена
1	до 20000	до 15000	до 2	30	70

4.9 Профилактика мышечных расстройств

Длительное пребывание в одном и том же положении и повторение одних и тех же движений вызывает различные мышечные расстройства.

Для профилактики возникновения мышечных расстройств при работе на ПК рекомендуется выполнение следующих требований:

- руки должны быть выпрямлены в запястьях и согнуты в локтях примерно под прямым углом, пальцы также должны быть слегка согнуты;

- удары по клавишам не должны быть слишком сильными;
 - рабочее кресло должно иметь подлокотники, отрегулированные соответствующим образом, которые служили бы опорой для рук как при работе с клавиатурой, так и при пользовании мышью;
 - при чувстве напряженности или спазмов в мышцах следует немедленно прекратить работу;
 - высота рабочего стула должна быть отрегулирована так, чтобы бедра были параллельны полу;
 - ноги должны твердо стоять на полу, а если приходится работать за высоким столом, следует воспользоваться подставкой;
 - сидеть нужно прямо или слегка подать корпус вперед, стараясь сохранять естественный изгиб тела в пояснице;
 - клавиатура и мышь должны быть расположены так, чтобы к ним не нужно было тянуться;
 - документ рекомендуется закреплять рядом с монитором специальной подставкой;
 - рекомендуется держать на столе эластичную резиновую игрушку или кольцо эспандер, и время от времени с его помощью разминать кисти рук;
 - через некоторое время менять режим работы.
- Соблюдение описанных выше требований поможет избежать неприятных последствий при работе на ПК.

4.10 Расчет микроклимата рабочей зоны

Микроклимат в рабочей зоне определяется действующими на организм человека сочетаниями температуры, влажности и скорости движения воздуха, а также температурой окружающих поверхностей. Повышенная влажность затрудняет теплоотдачу организма путем испарений при высокой температуре воздуха и способствует перегреву, а при низкой температуре, наоборот, усиливает теплоотдачу, способствуя переохлаждению. Оптимальны такие параметры микроклимата, которые при длительном и систематическом воздействии на человека обеспечивают сохранение нормального функционального и теплового состояния организма без напряжения реакций терморегуляции, что создает ощущение теплового комфорта и служит предпосылкой для высокой работоспособности. Поддержание оптимального микроклимата возможно только в том случае, если предприятие оснащено установкам кондиционирования микроклимата. В остальных случаях следует обеспечивать допустимые микроклиматические условия, т.е. такие, при которых хотя и могут возникать напряжения терморегуляции организма, но не выходят за пределы его физиологических приспособительных возможностей. При этом не возникают нарушения состояния здоровья, но может наблюдаться ухудшение самочувствия и понижение работоспособности.

Обеспечение воздушного комфорта в жилых, общественных и производственных помещениях зависит от систем аспирации, вентиляции и

кондиционирования воздуха. Воздушное отопление, совмещенное с вентиляцией, создает в помещении вполне удовлетворительный климат и обеспечивает благоприятные условия воздушной среды. Система кондиционирования воздуха, помимо выполнения задач вентиляции и отопления, позволяет создать благоприятный микроклимат в летний жаркий период года, благодаря использованию в своем составе фреоновой холодильной машины. Задача кондиционирования воздуха состоит в поддержании таких параметров воздушной среды, при которых каждый человек благодаря своей индивидуальной системе автоматической терморегуляции организма чувствовал бы себя комфортно, т.е. не замечал влияния этой среды.

Рассчитаем расход воздуха по формуле

$$L = (Q_{\text{изб}} * 860) / (C * \Delta t * \gamma),$$

$$Q_{\text{изб}} = Q_{\text{об}} + Q_{\text{oy}} + Q_{\text{рад}}$$

где $C=0,24$ ккал/(кг x °C) - теплоемкость воздуха;

$\gamma=1,206$ кг/м³ - удельная масса приточного воздуха.

$Q_{\text{об}} = P_{\text{об}} * \eta * n_{\text{об}}$ - количество тепла от оборудования.

$$Q_{\text{об}} = 10 * 0,5 * 4,75 \text{ кВт.}$$

$Q_{\text{oy}} = \eta N_{\text{oy}} * F$ - количество тепла от осветительных устройств.

где F - площадь помещения (м²).

$$Q_{\text{oy}} = 0,92 * 40 * 10 * 4 = 1,47 \text{ кВт}$$

Рассчитаем количество тепла от людей

$$Q_{\text{л}} = q_{\text{я}} * n$$

где $q_{\text{я}}$ - тепловыделение явное (Вт);

$Q_{\text{рал}}$ - количество тепла от солнечного излучения.

$$Q_{\text{рад}} = m * F * (q1 + q2) * B * k1 * k2$$

где m - количество окон;

F_0 - площадь окна;

B - коэффициент теплопропускания;

$k1$ - коэффициент затемнения остекления;

$k2$ - коэффициент загрязнения стекла.

$$Q_{\text{рад}} = 3 * 3 * (214 + 79 + 73 + 77) * 0,15 * 0,95 * 1,28 = 0,73, \text{ кВт.}$$

$$L = (4,75 + 1,47 + 0,5 + 0,73) * 860 / 0,24 * 1,206 * 8 = 2766,98 \text{ м}^3/\text{час.}$$

Итак за один восьмичасовой рабочий день расход воздуха , для одного программиста составит 2766,98 м³.

4.11 Расчет мер защиты от поражения электрическим током

Расчет заземления производится для того чтобы определить сопротивление сооружаемого контура заземления при эксплуатации, его размеры и форму. Как известно, контур заземления состоит из вертикальных заземлителей, горизонтальных заземлителей и заземляющего проводника. Вертикальные заземлители вбиваются в почву на определенную глубину. Горизонтальные заземлители соединяют между собой вертикальные заземлители. Заземляющий проводник соединяет контур заземления непосредственно с электрощитом. Размеры и количество этих заземлителей, расстояние между ними, удельное сопротивление грунта – все эти параметры напрямую зависят на сопротивление заземления. Заземление служит для снижения напряжения прикосновения до безопасной величины. Благодаря заземлению опасный потенциал уходит в землю тем самым, защищая человека от поражения электрическим током. Величина тока стекания в землю зависит от сопротивления заземляющего контура. Чем сопротивление будет меньше, тем величина опасного потенциала на корпусе поврежденной электроустановки будет меньше. Заземляющие устройства должны удовлетворять возложенным на них определенным требованиям, а именно величины сопротивление растекания токов и распределения опасного потенциала. Поэтому основной расчет защитного заземления сводится к определению сопротивления растекания тока заземлителя. Это сопротивление зависит от размеров и количества заземляющих проводников, расстояния между ними, глубины их заложения и проводимости грунта.

Работу необходимо проводить в соответствии с правилами технической эксплуатации электроустановок . Классификация помещений по опасности поражения электрическим током, по квалификации персонала и напряжению сетей. Также необходимы вводный и периодический инструктажи по технике безопасности при работе с электрическими приборами, соблюдать трудовую дисциплину, правильно организовать рабочее место, регулярно проходить медицинское обследование.

В лаборатории должен поддерживаться нормальный микроклимат, отсутствовать агрессивная среда, разъедающая изоляцию, вспомогательные материалы и оборудование должны быть не токопроводящими. Не токопроводящие полы. Различный цвет фаз. Шины защитного заземления расположены в местах, доступных для проверки и сохранности. Лаборатория должна быть обеспечена аптечкой.

Технические меры защиты. Для защиты от прикосновения к токоведущим частям оборудования и приборов используют изоляцию, недоступное расположение токоведущих частей и ограждение. Для защиты от поражения электрическим током при прикосновении к металлическим частям оборудования, которые могли случайно оказаться под напряжением, выполнено защитное заземление корпуса установки, $R < 4 \text{ Ом}$.

Сопротивление одиночного заземлителя определяется по формуле

$$R_{\text{оз}} = 0,366 \cdot \rho \cdot (\lg(2 \cdot L/d) + 0,5 \cdot \lg[(2 \cdot 4 \cdot t + L)/(4 \cdot t \cdot L)]) / L \quad (4.1)$$

где $R_{\text{оз}}$ - сопротивление одиночного заземлителя, Ом;

ρ - удельное сопротивление грунта, Ом.м;

L - длина заземлителя, м;

t - глубина заложения заземлителя, м;

d - диаметр заземлителя, м.

В используемой установке применяются заземлители из угловой стали, для которых определяется эквивалентный диаметр

$$d_{\text{экв}} = 0,95 \cdot b \quad (4.2)$$

где b - ширина сторон уголка, м.

$$d_{\text{экв}} = 0,95 \cdot 0,05,$$

$$d_{\text{экв}} = 0,0475 \text{ м}$$

Поэтому формула (4.1) для заземлителя из угловой стали примет вид:

$$R_{\text{оз}} = 0,366 \cdot \rho \cdot (\lg(2 \cdot L / d_{\text{экв}}) + 0,5 \cdot \lg((2 \cdot 4 \cdot t + L) / (2 \cdot 4 \cdot t \cdot L))) / L \quad (4.3)$$

$$R_{\text{оз}} = 0,366 \cdot 50 (\lg(2 \cdot 2,5 / 0,0475) + 0,5 \cdot \lg((2 \cdot 4 \cdot 70 + 2,5) / (2 \cdot 4 \cdot 70 \cdot 2,5))) / 2,5$$

Из формулы 4.3 следует

$$R_{\text{оз}} = 17,5 \text{ Ом}$$

Количество заземлителей определяется по формуле

$$n = R_{\text{оз}} / R_{\text{зн}} \quad (4.4)$$

где n - число заземлителей, шт.;

$R_{\text{оз}}$ - сопротивление одиночного заземлителя, Ом;

$R_{\text{зн}}$ - сопротивление заземлителя по нормам (4 Ом).

Расстояние между заземлителями рассчитывается по формуле

$$a = 2 * L \quad (4.5)$$

где a - расстояние между заземлителями, м;
 L - длина заземлителя, м.

$$a = 2 * 2,5 = 5 \text{ м}$$

Число заземлителей с учетом их взаимного экранирования определяется по формуле

$$n_э = n / \eta_{эз} \quad (4.6)$$

где $n_э$ - число заземлителей с учетом их взаимного экранирования, шт.;
 n - число заземлителей без учета их взаимного экранирования, шт.;
 $\eta_{эз}$ - коэффициент использования заземлителей учитывающий их взаимное экранирование.

$$n_э = 4 / 0,88,$$

$$n_э = 5 \text{ шт.}$$

Длина заземляющих проводников определяется по формуле

$$L_n = 1,05 * a * n_э \quad (4.7)$$

где L_n - длина заземляющих проводников, м;
 a - расстояние между заземлителями;
 $n_э$ - число заземлителей с учетом их взаимного экранирования.

$$L_n = 1,05 * 5 * 5; L_n = 26,25 \text{ м}$$

Сопротивление заземляющего проводника находится по формуле

$$R_{пн} = 0,36 * \rho * (\lg(2 * L_n / b * t)) / L \quad (4.8)$$

где $R_{пн}$ - сопротивление заземляющего проводника из полосовой стали, Ом;
 L_n - длина заземляющих проводников, м;
 b - ширина полосы заземляющего проводника, м;
 t - глубина заложения заземлителя, м.

$$R_{пн} = 0,366 * 50 * (\lg(2 * 26,25 / 0,05 * 0,7)) / 2,5,$$

$$R_{\Pi\Pi} = 3,89 \text{ Ом}$$

Действительное сопротивление заземляющего проводника с учетом взаимного экранирования заземлителей и заземляющих проводников находится по формуле

$$R_{\Pi\Pi} = R_{\Pi\Pi} / \eta_n \quad (4.9)$$

где $R_{\Pi\Pi}$ - действительное сопротивление заземляющего проводника, Ом;
 $R_{\Pi\Pi}$ - сопротивление заземляющего проводника, Ом;
 η_n - коэффициент использования заземляющего проводника.

$$R_{\Pi\Pi} = 3,89 / 0,89,$$

$$R_{\Pi\Pi} = 4,37 \text{ Ом}$$

Сопротивление растеканию тока всего заземляющего устройства находится по формуле

$$R_{зп} = R_{оз} R_{\Pi\Pi} / (R_{оз} \eta_n + R_{\Pi\Pi} \eta_{эз} n) \quad (4.10)$$

где $R_{зп}$ - сопротивление растеканию тока всего заземляющего устройства, Ом

$$R_{зп} = 17,5 * 4,37 / (17,5 * 0,89 + 4,37 * 0,88.4),$$

$$R_{зп} = 2,47 \text{ Ом}$$

Действительное число заземлителей определяется по формуле

$$n = R_{оз} / \eta_{эз} \cdot R_{зп} \quad (4.11)$$

где n - действительное число заземлителей, шт

$$n = 17,5 / 0,88.2,47 = 8 \text{ шт}$$

Итак, проектируемый заземлитель состоит из 8 вертикальных стержней из угловой стали длиной 2,5 м с эквивалентным диаметром 0,047 м, расположенных друг от друга на расстоянии 5 м и стального горизонтального электрода длиной 26,25 м, заглубленных в землю на 0,7 м. Общее сопротивление грунтового заземлителя 2,47 Ом, что обеспечивает безопасность работы на оборудовании.

Заключение

В данном дипломном проекте был разработан программный продукт для исследования времяпролетных свойств двумерной электронных линз. Целью разработки данного продукта являлось создание ПО, понятного и удобного для упрощения решения сложных вычислительных процессов, связанных с электронной оптикой.

Была изучена предметная область задачи, поставлены и решены определенные задачи, описаны основные требования к функциональности программного обеспечения, описаны требования к компьютеру, требования к программному обеспечению.

Для данного программного продукта были рассчитаны экономическая эффективность, был произведен расчет климата (воздухообмена), а также расчет заземления, влияющего на человека при работе с ним.

На сегодняшний день персональный компьютер дает возможность создавать продукты, позволяющие решать и выдавать ответы всего одним нажатием на кнопку, в то время как раньше на это уходил не один день. Участие ПК в современной жизни открывает новые возможности в помощи и решении старых и новых задач.

Список использованной литературы

- 1 З.Д. Еркешева, Г.Ш. Боканова. Методические указания к выполнению экономической части дипломных работ для студентов специальности 5В070400 - Вычислительная техника и программное обеспечение. - Алматы: АУЭС, 2013. - 40 с.
- 2 Экономика, организация и управление на предприятии Учебник/ Корсаков М.Н., Ребрин Ю.И., Федосова Т.В., Макареня Т.А., Шевченко И.К. и др.; Под ред. М.А.Боровской. - Таганрог: ТТИ ЮФУ, 2008. - 440с.
- 3 Агарков А. П. Системная организация и управление производством, его подготовкой и инфраструктурой: теория и практика: Монография. - М.: Хлебпродинформ, 2008. - 543 с.
- 4 Организация и планирование производства: Учеб. пособие / Под ред. А. Н. Ильченко, И. Д. Кузнецовой. - М.: Академия, 2008. - 208 с.
- 5 Скала Н.В. Основы организации охраны и безопасности труда в Республике Казахстан. - Алматы, 2007.
- 6 Безопасность жизнедеятельности. Белов С.В., Ильницкая А.В., Козьяков А.Ф. - 2007
- 7 Бимурзаев С.Б. Методы расчёта электронных линз: Учебное пособие. - Алматы, 2001. - 52 с.
- 8 Кельман В.М. и Явор С.Я. Электронная оптика. - Л.: Наука, 1968.
- 9 Баранова Л.А. и Явор С.Я. Электростатические электронные линзы. - М.: Наука, 1986.

Приложение А

```
unit Unit1;

interface

uses
    Windows, Messages, SysUtils, Variants, Classes, Graphics,
    Controls, Forms,
    Dialogs, StdCtrls, TeEngine, Series, ExtCtrls, TeeProcs,
    Chart;

type
    TForm1 = class(TForm)
        Label1: TLabel;
        Edit1: TEdit;
        Label2: TLabel;
        Label3: TLabel;
        Label4: TLabel;
        Label5: TLabel;
        Button1: TButton;
        Chart1: TChart;
        Series1: TFastLineSeries;
        Series2: TFastLineSeries;
        Series3: TFastLineSeries;
        procedure Button1Click(Sender: TObject);
    private
        { Private declarations }
    public
        { Public declarations }
    end;
    type Mass = array[0..2] of real;

var
    Form1: TForm1;

implementation

{$R *.dfm}

procedure TForm1.Button1Click(Sender: TObject);
var V2,h,ez,ez1,shz,chz,thz,thz1,thz2,
    z,zmin,zmax, fz,fz1,fz2, uz,uz1,uz2:real;
    i, n: integer;
    fm:mass;
begin

v2:=strtofloat(edit1.Text);

h := 0.1;
zmin := -3;
zmax := 3;
```

Продолжение Приложения А

```
z := zmin;

n := round((zmax-zmin)/h)+1;

label2.caption := '';
label3.caption := '';
label4.caption := '';
label5.caption := '';

series1.Clear;
series2.Clear;
series3.Clear;

for i:= 1 to n do begin

z := z + h;

ez:=exp(2.636*z);
ez1:=1/ez;
shz:=(ez-ez1)/2;
chz:=(ez+ez1)/2;
thz:=shz/chz;
thz1:=1/(chz*chz);
thz2:=-2*thz*thz1;

uz := arctan(z);
uz1 := 2.618*thz1;
uz2 := 2.618*thz2;

fm[0]:=0.5*((1+v2)+(v2-1)*uz);
fm[1]:=0.5*(v2-1)*uz1;
fm[2]:=0.5*(v2-1)*uz2;

label2.caption := label2.caption + floattostr(fm[0]) + '  ';
label3.caption := label3.caption + floattostr(fm[1]) + '  ';
label4.caption := label4.caption + floattostr(fm[2]) + '  ';
label5.caption := label5.caption + floattostr(z) + '  ';

series1.AddXY(z, fm[0], '');
series2.AddXY(z, fm[1], '');
series3.AddXY(z, fm[2], '');

end;
end;

end.

unit Unit1;

interface
```

uses

Продолжение Приложения А

Windows, Messages, SysUtils, Variants, Classes, Graphics,
Controls, Forms,
Dialogs, StdCtrls, ExtCtrls, TeeProcs, TeEngine, Chart,
Series, Grids;

type

```
TForm1 = class(TForm)
    Edit1: TEdit;
    Button1: TButton;
    Chart1: TChart;
    Series1: TFastLineSeries;
    Series2: TFastLineSeries;
    Series3: TFastLineSeries;
    Chart2: TChart;
    Series4: TFastLineSeries;
    Series5: TFastLineSeries;
    Chart3: TChart;
    Series6: TFastLineSeries;
    Series7: TFastLineSeries;
    Chart4: TChart;
    Label6: TLabel;
    Label11: TLabel;
    Button2: TButton;
    Button3: TButton;
    Button4: TButton;
    Series8: TFastLineSeries;
    Series9: TFastLineSeries;
    Series10: TFastLineSeries;
    Label2: TLabel;
    Label3: TLabel;
    Label4: TLabel;
    Label5: TLabel;
    Memo1: TMemo;
    Memo2: TMemo;
    Memo3: TMemo;
    Label7: TLabel;
    Label8: TLabel;
    Label9: TLabel;
    Label10: TLabel;
    Label11: TLabel;
    Label12: TLabel;

    Memo4: TMemo;
    Memo5: TMemo;
    procedure Button1Click(Sender: TObject);
    procedure Button2Click(Sender: TObject);
    procedure Button3Click(Sender: TObject);
    procedure Button4Click(Sender: TObject);
```


Продолжение Приложения А

```
    procedure Edit1Change(Sender: TObject);
    procedure Edit1KeyPress(Sender: TObject; var Key: Char);
private
    { Private declarations }
public
    { Public declarations }
end;
    type Mass = array[0..2] of real;
    procedure Pole(z, v2: Real; var FM: Mass);
    procedure Polesh(z1, v2: Real; var FMSH: Mass);
    procedure Runge(z, z1, v2: Real; var dm, em, gm, jm:
Mass);
    procedure Trajectory(r0, r01: Real; dm, gm: Mass; var R:
Mass);
var
    Form1: TForm1;
    z: Real;

implementation
uses Unit3, Unit2, Unit4;
{$R *.dfm}

    procedure Pole(z, v2: Real; var FM: Mass);
var ez, ez1, uz, uz1, uz2, shz, chz, thz, thz1, thz2, fz,
fz1, fz2 :Real;
begin

    ez := exp(2.618*z);
    ez1 := 1/ez;
    shz := (ez - ez1)/2;
    chz := (ez + ez1)/2;
    thz := shz/chz;
    thz1 := 1/(chz*chz);
    thz2 := (-2)*thz*thz1;

    uz := thz;
    uz1 := 2.618*thz1;
    uz2 := 2.618*thz2;

    fz := 0.5*((1 + v2) + (v2 - 1)*uz);
    fz1 := 0.5*(v2 - 1)*uz1;
    fz2 := 0.5*(v2 - 1)*uz2;
    FM[0] := fz;
    FM[1] := fz1;
    FM[2] := fz2;
end;
```

Продолжение Приложения А

```
procedure Polesh(z1, v2: Real; var fmsh: Mass);
var ezsh, ez1sh, uzsh, uz1sh, uz2sh, shzsh, chzsh,
    thzsh, thz1sh, thz2sh, fzsh, fzsh1, fzsh2 :Real;
begin

ezsh := exp(2.618*z1);
ez1sh := 1/ezsh;
shzsh := 0.5*(ezsh - ez1sh);
chzsh := 0.5*(ezsh + ez1sh);
thzsh := shzsh/chzsh;
thz1sh := 1/(chzsh*chzsh);
thz2sh := (-2)*thzsh*thz1sh;

uzsh := thzsh;
uz1sh := 2.618*thz1sh;
uz2sh := 2.618*thz2sh;

fzsh := 0.5*((1 + v2) + (v2 - 1)*uzsh);
fzsh1 := 0.5*(v2 - 1)*uz1sh;
fzsh2 := 0.5*(v2 - 1)*uz2sh;
FMsh[0] := fzsh;
FMsh[1] := fzsh1;
FMsh[2] := fzsh2;

end;

Procedure Runge(z, z1, v2: Real; var dm, em, gm, jm: Mass);
var dm2, em2, em3, dm3, em4, dm4,
    gm2, jm2, gm3, jm3, gm4, jm4,
    g1, g2, g3, g4,
    k1, k2, k3, k4,
    h, h2, zmin, zmax: real;
    i, n :integer;
    FM, FMSH: Mass;
begin

h := 0.01;
h2 := h/2;

Pole(z, v2, FM);
Polesh(z1, v2, FMSH);

k1 := em[1]*h;
dm2 := dm[0] + em[0]*h2;
em2 := em[0] + k1/2;

k2 := ((-1)/(4*FMSH[0]))*(2*FMSH[1]*em2 + FMSH[2]*dm2)*h;
```

Продолжение Приложения А

```
dm3 := dm[0] + em[0]*h2 + k1*h/4;
em3 := em[0] + k2/2;

k3 := ((-1)/(4*FMSH[0]))*(2*FMSH[1]*em3 + FMSH[2]*dm3)*h;
dm4 := dm[0] + em[0]*h + k2*h/4;
em4 := em[0] + k3;

k4 := ((-1)/(4*FM[0]))*(2*FM[1]*em4 + FM[2]*dm4)*h;

dm[0] := dm[0] + (em[0] + (k1 + k2 + k3)/6)*h;
em[0] := em[0] + (k1 + 2*k2 + 2*k3 + k4)/6;
em[1] := ((-1)/(4*FM[0]))*(2*FM[1]*em[0] + FM[2]*dm[0]);

g1 := jm[1]*h;
gm2 := gm[0] + jm[0]*h2;
jm2 := jm[0] + g1/2;

g2 := ((-1)/(4*FMSH[0]))*(2*FMSH[1]*jm2 + FMSH[2]*gm2)*h;
gm3 := gm[0] + jm[0]*h2 + g1*h/4;
jm3 := jm[0] + g2/2;

g3 := ((-1)/(4*FMSH[0]))*(2*FMSH[1]*jm3 + FMSH[2]*gm3)*h;
gm4 := gm[0] + jm[0]*h + g2*h/4;
jm4 := jm[0] + g3;

g4 := ((-1)/(4*FM[0]))*(2*FM[1]*jm4 + FM[2]*gm4)*h;

gm[0] := gm[0] + (jm[0] + (g1 + g2 + g3)/6)*h;
jm[0] := jm[0] + (g1 + 2*g2 + 2*g3 + g4)/6;
jm[1] := ((-1)/(4*FM[0]))*(2*FM[1]*jm[0] + FM[2]*gm[0]);

end;
procedure Trajectory (r0, r01: Real; dm, gm: Mass; var R:
Mass);
var z1, v2, sh: Real;
    jm, em : Mass;
begin
sh := 0.1;
Runge(z, z1, v2, dm, em, gm, jm);
R[0]:= r0*dm[0] + r01*gm[0];
end;

procedure TForm1.Button1Click(Sender: TObject);
var z, zmin, zmax, v2, h: real;
    i, n: integer;
    FM: Mass;
begin

v2 := strtofloat(edit1.Text);
```

Продолжение Приложения А

```
zmin := -3;
zmax := 3;
h := 0.01;
z := zmin;

n := round((zmax - zmin)/h) + 1;

series1.clear;
series2.clear;
series3.clear;
for i := 1 to n do
begin

z := z + h;
Pole (z, v2, FM);

series1.AddXY(z, FM[0], '');
series2.AddXY(z, FM[1], '');
series3.AddXY(z, FM[2], '');

memo1.lines.Add(floattostr(FM[0]));
memo2.lines.Add(floattostr(FM[1]));
memo3.lines.Add(floattostr(FM[2]));

end;
end;

procedure TForm1.Button2Click(Sender: TObject);
begin
form1.Close;
form3.Close;
form2.Close;
form4.Close;
end;

procedure TForm1.Button3Click(Sender: TObject);
var zmin, zmax, v2, h, h2, z, z1,
    k1, k2, k3, k4,
    g1, g2, g3, g4,
    dm2, em2, em3, dm3, em4, dm4,
    gm2, jm2, gm3, jm3, gm4, jm4: real;
    i, n: integer;
    FM, dm, em, gm, jm, FMSH : Mass;
begin

v2 := strtofloat(edit1.Text);

zmin := -3;
zmax := 3;
h := 0.01;
```

Продолжение Приложения А

```
h2 := h/2;
z := zmin;
z1 := zmin;

n := round((zmax - zmin)/h) + 1;

dm[0] := 1;
em[0] := 0;
dm[1] := em[0];
em[1] := 0;

gm[0] := 0;
jm[0] := 1;
gm[1] := jm[0];
jm[1] := 0;

series4.Clear;
series5.Clear;
series6.Clear;
series7.Clear;

for i := 1 to n do
begin

z := z + h;
z1 := z - h/2;

Pole (z, v2, FM);

Runge(z, z1, v2, dm, em, gm, jm);

series4.AddXY(z, dm[0], '');
series5.AddXY(z, em[0], '');

series6.AddXY(z, gm[0], '');
series7.AddXY(z, jm[0], '');

memo4.lines.add(floattostr(dm[0]));
memo5.Lines.add(floattostr(gm[0]));
end;
end;
procedure TForm1.Edit1KeyPress(Sender: TObject; var Key:
Char);
begin

if not (key in ['0'..'9', ',', '-', #8])
then key := #0;

end;
procedure TForm1.Edit1Change(Sender: TObject);
```

Продолжение Приложения А

```
begin

if (Edit1.Text <> '') then Button1.Enabled := True;
if (Edit1.Text <> '') then Button3.Enabled := True;
if (Edit1.Text <> '') then Button4.Enabled := True;

if (edit1.Text = '') then Button1.Enabled := False;
if (edit1.Text = '') then Button3.Enabled := False;
if (edit1.Text = '') then Button4.Enabled := False;

end;

end.

unit Unit2;

interface

uses
    Windows, Messages, SysUtils, Variants, Classes, Graphics,
Controls, Forms,
    Dialogs, StdCtrls, jpeg, ExtCtrls;

type
    TForm2 = class(TForm)
        Image1: TImage;
        Label1: TLabel;
        Button1: TButton;
        Button2: TButton;
        Button3: TButton;
        Button4: TButton;
        procedure Button1Click(Sender: TObject);
        procedure Button2Click(Sender: TObject);
        procedure Button3Click(Sender: TObject);
        procedure Button4Click(Sender: TObject);
    private
        { Private declarations }
    public
        { Public declarations }
    end;

var
    Form2: TForm2;

implementation

uses Unit3, Unit4, Unit5;

{$R *.dfm}
```

Продолжение Приложения А

```
procedure TForm2.Button1Click(Sender: TObject);
begin
  form2.Close;
end;

procedure TForm2.Button2Click(Sender: TObject);
form3.show;
form2.Visible := false;
end;

procedure TForm2.Button3Click(Sender: TObject);
begin
  form4.show;
form2.Visible := false;
end;

procedure TForm2.Button4Click(Sender: TObject);
begin
  form5.show;
end;

end.

unit Unit3;

interface

uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics,
Controls, Forms,
  Dialogs, StdCtrls;

type
  TForm3 = class(TForm)
    Label1: TLabel;
    Label2: TLabel;
    Button1: TButton;
    Button2: TButton;
    Button3: TButton;
    procedure Button2Click(Sender: TObject);
    procedure Button1Click(Sender: TObject);
    procedure Button3Click(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  Form3: TForm3;
```

Продолжение Приложения А

```
implementation

uses Unit2, Unit1;

{$R *.dfm}

procedure TForm3.Button2Click(Sender: TObject);
begin
form3.close;
form2.close;
end;

procedure TForm3.Button1Click(Sender: TObject);
begin
form2.Show;
form3.Visible := false;
end;

procedure TForm3.Button3Click(Sender: TObject);
begin
form1.show;
form3.Visible := false;
end;

end.

unit Unit4;

interface

uses
    Windows, Messages, SysUtils, Variants, Classes, Graphics,
Controls, Forms,
    Dialogs, StdCtrls;

type
    TForm4 = class(TForm)
        Mem1: TMemo;
        Button1: TButton;
        Button2: TButton;
        procedure Button1Click(Sender: TObject);
        procedure Button2Click(Sender: TObject);
    private
        { Private declarations }
    public
        { Public declarations }
    end;

var
    Form4: TForm4;
```


Продолжение Приложения А

```
implementation

uses Unit2;

procedure TForm4.Button1Click(Sender: TObject);
begin
form4.Close;
form2.Visible := true;
form2.show;
end;

procedure TForm4.Button2Click(Sender: TObject);
begin
form4.Close;
form2.Close;
end;

end.

unit Unit5;

interface

uses
    Windows, Messages, SysUtils, Variants, Classes, Graphics,
Controls, Forms,
    Dialogs, jpeg, ExtCtrls, StdCtrls;

type
    TForm5 = class(TForm)
        Label1: TLabel;
        Image1: TImage;
        Label2: TLabel;
        Label3: TLabel;
        Label4: TLabel;
        Button1: TButton;
        procedure Button1Click(Sender: TObject);
    private
        { Private declarations }
    public
        { Public declarations }
    end;

var
    Form5: TForm5;

implementation

uses Unit2;
```

Окончание Приложения А

```
procedure TForm5.Button1Click(Sender: TObject);  
begin  
  form5.close;  
  form2.show;  
end;  
  
end.
```