

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РЕСПУБЛИКИ КАЗАХСТАН

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 пространственной
фокусировки осесимметричной электромагнитной линзы

Специальность Вычислительная техника и программное обеспечение

Выполнил (а) Холодов И.А. ИТ-10-3
(Фамилия и инициалы) группа

Научный руководитель доктор физ.-мат. наук Билмурзаев С.Б.
(Фамилия и инициалы, ученая степень, звание)

Консультанты:

по экономической части:

Бреева З.Д. с.с. преподаватель
(Фамилия и инициалы, ученая степень, звание)
Бреева « 13 » 05 2014 г.
(подпись)

по безопасности жизнедеятельности:

Трабевро Н.Б. Д.т.н., профессор
(Фамилия и инициалы, ученая степень, звание)
Трабевро « 14 » 05 2014 г.
(подпись)

по применению вычислительной техники:

Билмурзаев С.Б. доктор физ.-мат. наук
(Фамилия и инициалы, ученая степень, звание)
Билмурзаев « 24 » 05 2014 г.
(подпись)

(Фамилия и инициалы, ученая степень, звание)

« _____ » _____ 20__ г.

(подпись)

Нормоконтролер: Тусенов Д.М., ассистент
(Фамилия и инициалы, ученая степень, звание)
Тусенов « 23 » 05 2014 г.
(подпись)

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

Алматы 2014 г.

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РЕСПУБЛИКИ КАЗАХСТАН

Некоммерческое акционерное общество
АЛМАТИНСКИЙ УНИВЕРСИТЕТ ЭНЕРГЕТИКИ И СВЯЗИ

Факультет Информационных технологий
Специальность Вычислительная техника и программное обеспечение
Кафедра Компьютерных технологий

ЗАДАНИЕ

на выполнение дипломного проекта

Студент Холодов Илья Александрович
(фамилия, имя, отчество)

Тема проекта „Исследование свойств пространственной
фокусировки осесимметричной электронной линзы.“

утверждена приказом ректора № 115 от «24» сентября 2014 г.

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

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

Исходные данные объекта - свойства электронной
линзы, описанные ранее в основной литературе,
используемой при написании дипломного
проекта

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

- вычисление формул распределения поле электронной осесимметричной линзы;
- расчет электростатического распределения потенциала и его производных до второго порядка;
- получение уравнений траектории заряженных частиц с учетом величин до второго порядка малости;
- получение формул для определения параксиальных свойств и краевых аберраций электронной осесимметричной линзы;
- изучение численных методов для решения дифференциальных уравнений и определенных интегралов;
- разработка программного продукта для исследования фокусирующих свойств осесимметричной линзы.

Г Р А Ф И К
подготовки дипломного проекта

№ п/п	Наименование разделов, перечень разрабатываемых вопросов	Сроки представления руководителю	Примечание
1	Получение задания к дипломной работе	24.02.2014г	
2	Утверждение основной литературы	26.02.2014г	
3	Написание постановки задания к дипломной работе	28.02.2014г	
4	Выбор языка программирования	04.03.2014г	
5	Начало изложение теории чисел	11.03.2014г	
6	Выведение формулы распределения	14.03.2014г	
7	Расчет электростатического распределения потенциала	17.03.2014г	
8	Получение уравнения траектории заряженной частицы	18.03.2014г	
9	Изучение свойств парамагнетиков и ферромагнетиков	01.04.2014г	
10	Изучение численных методов решения дифференциальных уравнений и определенных интегралов	03.04.2014г	
11	Построение графиков парабол и траекторий	12.04.2014г	
12	Разработка программного продукта	15.04.2014г	
13	Отладка и тестирование ПП	24.05.2014г	

Дата выдачи задания « 24 » февраля 2014 г.

Заведующий кафедрой _____
(подпись) (Фамилия и инициалы)

Руководитель _____
(подпись) Бишурзаев С.Б.
(Фамилия и инициалы)

Задание принял к исполнению студент _____
(подпись) Колотов И.А.
(Фамилия и инициалы)

Андатпа

Осы диплом жұмыста «Осесимметриялық электрондық дүрбінің кеңістік фокустандыру ерекшеліктерін зерттеу» атындағы ПП ұсынған. ПП Borland Delphi 7 тілінде жазылған. Осы ПП-нің еңгізу нысанасы - электрондық оптика саласында физика есептерін шығаратын жеткіліксіз программалық база. Бұл программа электрондық бөлшектердің аяулы фокусировка және электрондық дала өлшеуі сұрақтары көтеріледі.

ҚТӘ жағынан программисттің жұмыс бөлмесінде Компьютерден электромагниттік сәуле шығаруы және өрт қауіпсіздігі санаулымдары өткізілген. Экономикалық жағынан ПП-нің тиімділігі және өзектілігі саналған.

Аннотация

В данном дипломном проекте представлен программный продукт под названием «Исследование свойств пространственной фокусировки осесимметричной электронной линзы».

Программный продукт разработан в программной среде Borland Delphi 7. Целью внедрения такого программного продукта является недостаточная программная база по решению физических задач в области электронной оптики. Данная программа затрагивает вопросы пространственной фокусировки частиц, всевозможные расчёты полей и траекторий прохождения частиц через двухэлектродные системы.

Также был произведён инженерный расчёт электромагнитного излучения от ЭВМ и расчёт противопожарной безопасности.

Произведён расчёт экономической эффективности разработанного программного продукта.

Annotation

In this diploma project the program product named “Researching of the properties of the dimensional focusing of axisymmetric electronic lens” has been represented.

Introduction purpose of this program product is insufficient program base of solving physical problems in electronic optics field. This program affects problems of particles’ dimensional focusing and omnifarious calculations of electric fields and trajectories of particles’ passages through two-electrode systems.

Содержание

Введение.....	12
1 Аналитическая часть проекта	14
1.1 Описание предметной области	14
1.2 Обзор аналогов	14
1.3 Постановка задачи.....	14
1.4 Цель дипломного проекта	15
1.5 Цель и назначение программного обеспечения.....	15
1.6 Актуальность программного обеспечения	15
1.7 Выбор языка и средства программирования.....	16
1.8 Требования к программному обеспечению.....	16
2 Основная часть	17
2.1 Понятие электронной линзы	17
2.2 Расчёт поля.....	18
2.2.1 Уравнение Лапласа	18
2.2.2 Общее представление ряда для пространственного распределения электростатического потенциала через его осевое распределение и отклонения частиц от оси симметрии.....	19
2.3 Уравнения движения.....	20
2.4 Уравнения траекторий с учётом величин малости до второго порядка.....	22
2.5 Электронно-оптические свойства.....	24
2.5.1 Параксиальное приближение	24
2.5.2 Определение коэффициентов хроматических аберраций.....	26
2.6 Используемые в работе численные методы	27
2.6.1 Метод Рунге-Кутты четвёртого порядка	27
2.6.2 Метод Симпсона для численного определения интегралов	28
2.7 Руководство пользователя.....	28
3 Техничко-экономическое обоснование проекта	46
3.1 Описание работы и обоснование необходимости	46
3.2 Трудовые ресурсы, используемые в работе	46
3.3 Расчёт стоимости работы по проектированию и разработке	47
3.4 Расчёт затрат на разработку программного обеспечения	53
3.5 Цена интеллектуального труда.....	56
3.6 Вывод.....	57
4 Безопасность жизнедеятельности и труда человека.....	58
4.1 Анализ вредных факторов, влияющих на здоровье программиста	58
4.2 Микроклимат рабочей зоны программиста.....	59
4.3 Освещение рабочего места.....	60
4.4 Воздействие электромагнитных излучений и электрического поля. Статическое электричество	61
4.5 Шум и вибрация	62
4.6 Другие вредные факторы, оказывающие воздействие на здоровье программиста.....	63

4.7 Характеристики рабочего помещения	64
4.8 Используемое оборудование.....	64
4.9 Расчётная часть.....	64
4.9.1 Инженерный расчёт электромагнитного излучения от компьютера.....	64
4.9.2 Расчёт противопожарной безопасности	66
4.10 Вывод.....	67
Заключение	68
Список использованной литературы.....	69
Приложение А	70

Введение

Электронная линза – устройство, создающее магнитное или электрическое поле для фокусировки электронных пучков, их формирования и получения электронно-оптических изображений. Электронные линзы классифицируют по типу поля на магнитные и электростатические и по виду симметрии на осесимметричные, цилиндрические, квадрупольные и другие.

Электростатические осесимметричные линзы подразделяются на иммерсионные, одиночные и катодные. Осесимметричные линзы могут состоять из нескольких электродов различной формы, находящихся под разными потенциалами. Это диафрагмы с круглыми отверстиями, полые цилиндры, конусы и т. п. Простейшей линзой является одиночная диафрагма, поле которой с одной или с двух сторон граничит с однородными электрическими полями. В зависимости от приложенного к диафрагме потенциала и направления примыкающих полей она может быть как собирающей, так и рассеивающей.

Электронные линзы составляют основную часть большинства электронно-оптических приборов. История электронных линз начинается в 1926 году. В этом году Буш опубликовал работу, в которой было показано, что электрические и магнитные поля, обладающие вращательной симметрией, фокусируют узкие пучки заряженных частиц, т.е. действуют как линзы. Дальнейшее развитие электронной оптики привело к подробной разработке теории и конструкций как электростатических, так и магнитных электронных линз. Выбор между ними при практическом использовании определяется возможностью наиболее полного удовлетворения требованиям, выдвигаемым конкретной задачей. Каждый из этих видов линз обладает и достоинствами, и недостатками. К достоинствам электростатических линз следует отнести их меньший вес, размеры, отсутствие потребления мощности питания, что облегчает стабилизацию и уменьшает вес источников напряжения, большую простоту изготовления. Они намного менее инерционны, чем магнитные, и поэтому используются при работе с быстропротекающими процессами. К достоинствам магнитных линз можно отнести меньшие aberrации по сравнению с электростатическими и лёгкость их установления вне вакуумной камеры.

Главная цель для разработки компьютерной программы для изучения свойств пространственной фокусировки осесимметричной электронной линзы заключается в изучении свойств пространственной фокусировки осесимметричных электронных линз.

Разработка данного программного продукта позволит производить физические исследования осесимметричных электронных линз с минимальными затратами времени. Отсутствие аналогов программ, связанных с данной тематикой, делает данный программный продукт актуальным.

Объектом исследования в данном дипломном проекте является система, состоящая из двухэлектродной осесимметричной электронной линзы. Сама по себе электронная оптика является достаточно новой наукой, соответственно и все программные продукты являются актуальными на данный момент. Научная проблема в создании программного продукта для быстрого решения систем электронных линз пока остается открытой, но создание программного продукта облегчит решение задач такого плана.

Целью данного дипломного проекта является разработка программного продукта для расчёта всевозможных систем осесимметричных электронных линз.

1 Аналитическая часть проекта

1.1 Описание предметной области

Электронные линзы – устройства, создающие магнитное или электрическое поле для фокусировки электронных пучков, их формирования и получения электронно-оптических изображений. Их классифицируют по типу поля (магнитные, электростатические), по виду симметрии (осесимметричные, цилиндрические, квадрупольные и др.).

Электростатические осесимметричные линзы делятся на иммерсионные, одиночные и катодные. Они состоят из нескольких электродов различной формы, находящихся под разными потенциалами. Это диафрагмы с круглыми отверстиями, полые цилиндры, конусы и т. п. Одной из простейших линз является линза, состоящая из трубчатых цилиндров.

1.2 Обзор аналогов

В настоящее время имеется достаточное количество программных продуктов, написанных на различных языках программирования, направленных на решение физических задач различной сложности и направленности. Причём есть как программные продукты, разработанные профессиональными программистами и являющиеся коммерческими продуктами, так и программы, написанные любителями и направленные на решение локальных задач.

Одной из таких программ является программа MathCad. MathCad – система компьютерной алгебры из класса систем автоматизированного проектирования, ориентированная на подготовку интерактивных документов с вычислениями и визуальным сопровождением, отличается лёгкостью использования и применения для коллективной работы. MathCad был написан А.Раздовом из Массачусетского технологического института (MIT).

Несмотря на то, что MathCad не ориентирован на пользователей-программистов, он также используется в сложных проектах для визуализации результатов математического моделирования путём использования распределённых вычислений и традиционных языков программирования. Также Mathcad используется в крупных инженерных проектах, где большое значение имеет соответствие стандартам. Mathcad очень удобно использовать для обучения, вычислений и инженерных расчетов.

1.3 Постановка задачи

Главная цель для разработки компьютерной программы для изучения свойств пространственной фокусировки осесимметричной электронной линзы заключается в расчёте поля и коэффициентов по методу Рунге-Кутты для построения траектории частицы, находящейся в этом поле.

Задача программы сводится к следующим этапам компьютерных операций:

- получение формулы для распределения поля электронной линзы с вращательной симметрией на основании уравнения Лапласа;
- расчёт осевого распределения электростатического потенциала и его производных до второго порядка;
- получение уравнений траекторий для пучков заряженных частиц с учётом величин до второго порядка малости;
- получение формул, определяющих параксиальные свойства и хроматические аберрации электронной линзы;
- ознакомление с методом Рунге-Кутты для численного решения дифференциальных уравнений;
- ознакомление с методом Симпсона для численного определения определённых интегралов;
- создание программного продукта для исследования фокусирующих свойств двухэлектродной электронной линзы с вращательной симметрией;
- нахождение частных решений параксиального уравнения траекторий методом Рунге-Кутты четвёртого порядка;
- отладка, тестирование и оптимизация программного продукта;
- определение кардинальных элементов и коэффициентов аберраций;
- построение таблиц и графиков зависимостей от распределения потенциала траекторий частиц, кардинальных элементов и коэффициентов аберраций.

1.4 Цель дипломного проекта

Целью дипломного проекта является разработка программного продукта для исследования фокусирующих свойств осесимметричных электронных линз.

1.5 Цель и назначение программного обеспечения

Целью программного обеспечения реализация возможности исследования осесимметричных линз и движения частиц в них.

Назначением программного обеспечения является упрощение решения трудоёмких вычислений, связанных с электронной оптикой.

1.6 Актуальность программного обеспечения

Разработка данного программного продукта позволит производить физические исследования осесимметричных электронных линз с минимальными затратами времени. Новизна предмета исследования, связанная с данной тематикой, делает данный программный продукт актуальным.

1.7 Выбор языка и средства программирования

Для разработки программного продукта был использован язык Pascal, для реализации проекта – программная среда разработки Borland Delphi 7.

Pascal является языком программирования общего назначения, одним из наиболее известных языков программирования, а также базой для ряда других языков.

Строгая типизация и наличие средств структурного (процедурного) программирования являются главными особенностями Pascal. Pascal был одним из первых таких языков. По настоянию Вирта наряду со строгой типизацией, в Паскале сведены к минимуму возможные синтаксические неоднозначности, а сам синтаксис разработчики попытались сделать максимально понятным даже при первом знакомстве с языком.

Тем не менее, первоначально язык не имел возможности передачи функциям массивов переменной длины, в языке отсутствовали средства работы с динамической памятью, была ограниченная библиотека ввода-вывода, отсутствовали средства для подключения функций, написанных на других языках, отсутствовали средства отдельной компиляции и т.п. Подробный разбор недостатков языка Паскаль того времени был выполнен Б.Кёрниганом в статье «Почему Pascal не является моим любимым языком программирования». Некоторые недостатки Паскаля были исправлены в 1982 году: в языке появились открытые массивы, давшие возможность использовать одни и те же процедуры для обработки одномерных массивов различных размеров.

Диалекты Паскаля, применяемые в Turbo Pascal для DOS и Delphi для Windows, стали популярны из-за отсутствия других успешных коммерческих реализаций.

Delphi – объектно-ориентированный язык программирования, являющийся диалектом языка Pascal. Компания Borland, начиная со среды разработки Delphi 7.0, в официальных документах стала использовать название «Delphi» для обозначения языка Pascal. Начиная с 2007 Delphi начал жить как отдельный язык своей самостоятельной жизнью и претерпевал различные изменения, направленные на оптимизацию некоторых вещей. К примеру, появились class helpers, перегрузки операторов и другое.

1.8 Требования к программному обеспечению

- ОС Microsoft Windows XP;
- процессор Intel Core 2 Duo 2.0 GHz;
- оперативная память: 1 GB RAM;
- видеоадаптер: NVIDIA GeForce 450M 512 MB;
- жёсткий диск: 1 GB свободного места на HDD.

2 Основная часть

2.1 Понятие электронной линзы

Электронные линзы составляют основную часть большинства электронно-оптических приборов. История электронных линз начинается в 1926 году. В этом году Буш опубликовал работу, в которой было показано, что электрические и магнитные поля, обладающие вращательной симметрией, фокусируют узкие пучки заряженных частиц, т.е. действуют как линзы. Дальнейшее развитие электронной оптики привело к подробной разработке теории и конструкций как электростатических, так и магнитных электронных линз. Выбор между ними при практическом использовании определяется возможностью наиболее полного удовлетворения требованиям, выдвигаемым конкретной задачей. Каждый из этих видов линз обладает и достоинствами, и недостатками. К достоинствам электростатических линз следует отнести их меньший вес, габариты, отсутствие потребления мощности питания, что облегчает стабилизацию и уменьшает вес источников напряжения, большую простоту изготовления. Они гораздо менее инерционны, чем магнитные, и поэтому используются при работе с быстропротекающими процессами. К достоинствам магнитных линз можно отнести меньшие aberrации по сравнению с электростатическими линзами, а также лёгкость их установления вне вакуумной камеры.

Электронные линзы данного проекта – осесимметричные, они состоят из двух электродов, представленных в виде двух цилиндров с круглыми отверстиями, имеющих одинаковые диаметры (рисунок 2.1).

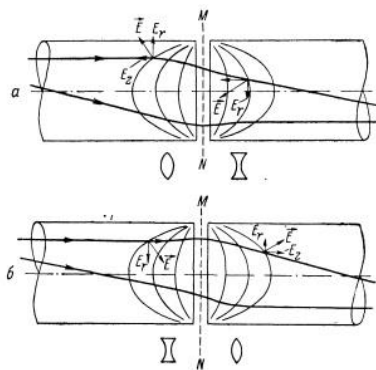


Рисунок 2.1 – Иммерсионная линза

Здесь V_1 – потенциал первого (со стороны падающего луча) цилиндра, V_2 – второго. Для удобства расчёта значения потенциалов на электродах задаются в единицах V_1 , т.е. $\frac{V_2}{V_1}$. Могут быть два случая: $\frac{V_2}{V_1} > 1$ и $\frac{V_2}{V_1} < 1$. В первом случае электроны, проходя линзу, будут ускоряться, во втором – замедляться.

Электрическое поле сосредоточено вблизи обращённых друг к другу концов цилиндров.

2.2 Расчёт поля

Расчёт любой системы в электронной оптике начинается с расчёта поля. Для электронных линз с вращательной симметрией целесообразно ввести цилиндрическую систему координат r, z, ψ .

В цилиндрической системе координат r, z, ψ потенциал φ записывается в виде

$$\varphi = \varphi(r, z, \psi) \quad (2.1)$$

Если поле обладает вращательной симметрией, то потенциал не зависит от угловой координаты ψ , т.е. $\frac{\partial \varphi}{\partial \psi} = 0$, значит $\varphi = \varphi(r, z)$.

Потенциал в глубине каждого цилиндра постоянен. Таким образом, электрическое поле линзы здесь граничит с двумя областями, в которых потенциалы постоянны, но различны по величине, как это и требуется по определению иммерсионной линзы.

2.2.1 Уравнение Лапласа

Распределение потенциалов любой электронно-оптической системы подчиняется уравнению Лапласа.

Уравнение Лапласа – дифференциальное уравнение в частных производных. В прямоугольной системе координат уравнение Лапласа записывается так

$$\Delta \varphi = \frac{\partial^2 \varphi}{\partial x^2} + \frac{\partial^2 \varphi}{\partial y^2} + \frac{\partial^2 \varphi}{\partial z^2} = 0 \quad (2.2)$$

Чтобы получить уравнение Лапласа в цилиндрической системе координат, воспользуемся соотношениями

$$x = r \cos \psi, \quad y = r \sin \psi, \quad r = \sqrt{x^2 + y^2} \quad (2.3)$$

В этом случае имеют место соотношения

$$\frac{\partial \varphi}{\partial x} = \frac{\partial \varphi}{\partial r} \frac{\partial r}{\partial x}, \quad (2.4)$$

$$\frac{\partial^2 \varphi}{\partial x^2} = \frac{\partial}{\partial x} \left(\frac{\partial \varphi}{\partial x} \right) = \frac{\partial}{\partial x} \left(\frac{\partial \varphi}{\partial r} \frac{\partial r}{\partial x} \right) = \frac{\partial \varphi}{\partial r} \frac{\partial^2 r}{\partial x^2} + \left(\frac{\partial r}{\partial x} \right)^2 \frac{\partial^2 \varphi}{\partial r^2}, \quad (2.5)$$

$$\frac{\partial \varphi}{\partial y} = \frac{\partial \varphi}{\partial r} \frac{\partial r}{\partial y}, \quad (2.6)$$

$$\frac{\partial^2 \varphi}{\partial y^2} = \frac{\partial}{\partial y} \left(\frac{\partial \varphi}{\partial y} \right) = \frac{\partial}{\partial y} \left(\frac{\partial \varphi}{\partial r} \frac{\partial r}{\partial y} \right) = \frac{\partial \varphi}{\partial r} \frac{\partial^2 r}{\partial y^2} + \left(\frac{\partial r}{\partial y} \right)^2 \frac{\partial^2 \varphi}{\partial r^2} \quad (2.7)$$

Выразим $\frac{\partial^2 r}{\partial x^2}$ и $\frac{\partial^2 r}{\partial y^2}$

$$\frac{\partial r}{\partial x} = \frac{2x}{2\sqrt{x^2+y^2}} = \frac{x}{\sqrt{x^2+y^2}}, \quad (2.8)$$

$$\frac{\partial^2 r}{\partial x^2} = \frac{\partial}{\partial x} \left(\frac{\partial r}{\partial x} \right) = \frac{\partial}{\partial x} \left(\frac{x}{\sqrt{x^2+y^2}} \right) = \frac{x^2}{(x^2+y^2)^{3/2}}, \quad (2.9)$$

$$\frac{\partial r}{\partial y} = \frac{2y}{2\sqrt{x^2+y^2}} = \frac{y}{\sqrt{x^2+y^2}}, \quad (2.10)$$

$$\frac{\partial^2 r}{\partial y^2} = \frac{\partial}{\partial y} \left(\frac{\partial r}{\partial y} \right) = \frac{\partial}{\partial y} \left(\frac{y}{\sqrt{x^2+y^2}} \right) = \frac{y^2}{(x^2+y^2)^{3/2}} \quad (2.11)$$

Подставим полученные результаты в (2.1)

$$\begin{aligned} \Delta \varphi &= \frac{\partial^2 \varphi}{\partial x^2} + \frac{\partial^2 \varphi}{\partial y^2} + \frac{\partial^2 \varphi}{\partial z^2} = \frac{\partial \varphi}{\partial r} \frac{\partial^2 r}{\partial x^2} + \left(\frac{\partial r}{\partial x} \right)^2 \frac{\partial^2 \varphi}{\partial r^2} + \frac{\partial \varphi}{\partial r} \frac{\partial^2 r}{\partial y^2} + \left(\frac{\partial r}{\partial y} \right)^2 \frac{\partial^2 \varphi}{\partial r^2} + \\ &+ \frac{\partial^2 \varphi}{\partial z^2} = \frac{\partial^2 \varphi}{\partial r^2} \left[\left(\frac{\partial r}{\partial x} \right)^2 + \left(\frac{\partial r}{\partial y} \right)^2 \right] + \frac{\partial \varphi}{\partial r} \left(\frac{\partial^2 r}{\partial x^2} + \frac{\partial^2 r}{\partial y^2} \right) + \frac{\partial^2 \varphi}{\partial z^2} = \frac{\partial^2 \varphi}{\partial r^2} \left[\frac{x^2}{x^2+y^2} + \frac{y^2}{x^2+y^2} \right] + \\ &+ \frac{\partial \varphi}{\partial r} \left(\frac{x^2}{(x^2+y^2)^{3/2}} + \frac{y^2}{(x^2+y^2)^{3/2}} \right) + \frac{\partial^2 \varphi}{\partial z^2}, \quad (2.12) \end{aligned}$$

$$\Delta \varphi = \frac{\partial^2 \varphi}{\partial r^2} + \frac{1}{r} \frac{\partial \varphi}{\partial r} + \frac{\partial^2 \varphi}{\partial z^2} = 0 \quad (2.13)$$

2.2.2 Общее представление ряда для пространственного распределения электростатического потенциала через его осевое распределение и отклонения частиц от оси симметрии

Путём некоторых преобразований можно переписать (2.13) в виде

$$\frac{\partial}{\partial r} \left(r \frac{\partial \varphi}{\partial r} \right) + \frac{\partial}{\partial z} \left(r \frac{\partial \varphi}{\partial z} \right) = 0 \quad (2.14)$$

После деления на r и некоторого преобразования (2.14) можно записать в виде

$$\frac{4}{2r} \frac{\partial}{\partial r} \left(\frac{r^2}{2r} \frac{\partial \varphi}{\partial r} \right) + \frac{\partial^2 \varphi}{\partial z^2} = 0 \quad (2.15)$$

Если положить $r^2 = u$, то

$$4 \frac{\partial}{\partial u} \left(u \frac{\partial \varphi}{\partial u} \right) + \frac{\partial^2 \varphi}{\partial z^2} = 0 \quad (2.16)$$

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

$$\varphi(r, z) = \Phi(z) + r^2 \Phi_2(z) + r^4 \Phi_4(z) + \dots \quad (2.17)$$

где $\Phi(z) = \varphi(0, z)$ обозначает осевое распределение потенциала, т.е. для $r=0$. Если ряд, представленный в формуле (2.17), подставить в уравнение Лапласа (2.13), то, сравнивая коэффициенты при одинаковых степенях r , получим значения функций Φ_2, Φ_4, \dots так, что $\varphi(r, z)$ примет вид

$$\varphi(r, z) = \Phi(z) - \frac{r^2}{4} \Phi''(z) + \frac{r^4}{64} \Phi^{IV}(z) - \dots \quad (2.18)$$

Для целей электронной оптики достаточно знать первые три члена, записанные в (2.18).

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

$$\Phi(z) = \frac{1}{2} [(V_1 + V_2) + (V_2 - V_1)U(z)] \quad (2.20)$$

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

$$U(z) = th \left(\frac{2,636z}{d} \right) \quad (2.21)$$

2.3 Уравнения движения

Уравнения движения частицы массы m и заряда q имеют вид

$$m\ddot{r} = -q \frac{\partial \varphi}{\partial r}, \quad (2.22)$$

$$\frac{m}{2} \left[\left(\frac{dz}{dt} \right)^2 + \left(\frac{dr}{dt} \right)^2 \right] = -q(\varphi + \varepsilon) \quad (2.23)$$

где ε – величина, характеризующая разброс по энергиям частиц.

В отличие от механики, где изучается движение отдельной материальной точки, в электронной оптике рассматривают семейство траекторий. Поэтому, необходимо из уравнений движения перейти к уравнениям траекторий, путём исключения времени и перехода к зависимости от координаты z оптической оси, Для этого воспользуемся следующими соотношениями

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

$$\dot{r} = \frac{dr}{dt} = \dot{z} \frac{dr}{dz} = \dot{z} r',$$

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

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

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

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

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

$$\ddot{r} = \frac{d}{dt}(\dot{r}) = \dot{z} \frac{d}{dz}(\dot{z} r') = \dot{z} \left(\frac{d\dot{z}}{dz} r' + \dot{z} \frac{d}{dz} r' \right) = \frac{r'}{2} \frac{d}{dz}(\dot{z}^2) + \dot{z}^2 r'' \quad (2.23)$$

Тогда

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

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

$$\ddot{r} = k^2 \left[\frac{r'}{2} \frac{d\Phi(z)}{dz} + \Phi(z) r'' \right] \quad (2.24)$$

Из (2.22) известно, что

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

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

$$\frac{\partial \varphi}{\partial r} = -\frac{r}{2} \Phi''(z) + \frac{r^3}{16} \Phi^{IV}(z) + \dots \quad (2.25)$$

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

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

Отсюда следует, что

$$\frac{d^2 r}{dz^2} + \frac{\Phi'}{2\Phi} \frac{dr}{dz} + \frac{\Phi''}{4\Phi} r = 0 \quad (2.28)$$

Уравнение (2.28) можно записать и так

$$\Phi r'' + \frac{\Phi' r'}{2} + \frac{\Phi'' r}{4} = 0 \quad (2.29)$$

Таким образом, для удаления электрона от оси получилось линейное дифференциальное уравнение второго порядка. Оно было впервые получено Г.Бушем в 1926 году. Уравнение (2.29) является линейным однородным дифференциальным уравнением второго порядка. Из теории известно, что общее решение такого уравнения есть суперпозиция двух его линейно независимых частных решений

$$r = ap(z) + bg(z) \quad (2.30)$$

Здесь a и b – произвольные постоянные, а $p=p(z)$ и $g=g(z)$ – аналитические функции.

2.4 Уравнения траекторий с учётом величин малости до второго порядка

Чтобы рассчитать хроматические aberrации, нужно в уравнение параксиальных траекторий вместо $\Phi(z)$ подставить $\Phi(z) + \varepsilon$. Получим

$$(\Phi + \varepsilon)r'' + \frac{\Phi r'}{2} + \frac{\Phi'' r}{4} = 0 \quad (2.31)$$

Значит

$$\Phi r'' + \frac{\Phi r'}{2} + \frac{\Phi'' r}{4} = -\varepsilon r'' \quad (2.32)$$

Уравнение (2.32) – линейное неоднородное дифференциальное уравнение второго порядка. Известно, что общее решение линейного неоднородного уравнения второго порядка равно сумме общего решения соответствующего ему линейного уравнения и какого-либо частного решения нелинейного уравнения.

С учетом (2.30), общее решение уравнения (2.32) можно записать в виде

$$r = ap(z) + bg(z) + \chi(z) \quad (2.33)$$

где χ – частное решение уравнения (2.32).

Подставим (2.33) в (2.32) при условии, что

$$r' = ap'(z) + bg'(z) + \chi'(z),$$

$$r'' = ap''(z) + bg''(z) + \chi''(z)$$

Получим

$$\Phi \chi'' + \frac{\Phi'}{2} \chi' + \frac{\Phi''}{4} \chi = -\varepsilon r''$$

Используя метод вариации произвольных постоянных, получаем

$$\chi(z) = \frac{\varepsilon}{W} \left[p(z) \int_{z_0}^z \frac{g(z)r''}{\sqrt{\Phi}} dz - g(z) \int_{z_0}^z \frac{p(z)r''}{\sqrt{\Phi}} dz \right] \quad (2.34)$$

Представим χ в следующем виде

$$\chi = \varepsilon(a\chi_1 + b\chi_2) \quad (2.35)$$

Тогда из (2.34) следует

$$\chi_1 = \frac{1}{W} \left(p \int_{z_0}^z \frac{gp''}{\sqrt{\Phi}} - g \int_{z_0}^z \frac{pp''}{\sqrt{\Phi}} \right) = pJ_2 - gJ_1,$$

$$\chi_2 = \frac{1}{W} \left(p \int_{z_0}^z \frac{gg''}{\sqrt{\Phi}} - g \int_{z_0}^z \frac{pg''}{\sqrt{\Phi}} \right) = pJ_4 - gJ_3$$

где

$$J_1 = \frac{1}{W} \int_{z_0}^z \frac{pp''}{\sqrt{\Phi}} dz, \quad J_2 = \frac{1}{W} \int_{z_0}^z \frac{gp''}{\sqrt{\Phi}} dz, \quad (2.36)$$

$$J_3 = \frac{1}{W} \int_{z_0}^z \frac{pg''}{\sqrt{\Phi}} dz, \quad J_4 = \frac{1}{W} \int_{z_0}^z \frac{gg''}{\sqrt{\Phi}} dz$$

Окончательно уравнение траекторий можно записать в виде

$$r = r_0 + \Delta r \quad (2.37)$$

где

$$r_0 = ap(z) + bg(z) \quad (2.38)$$

– уравнение параксиальной траектории, а

$$\Delta r = \varepsilon(a\chi_1 + b\chi_2) \quad (2.39)$$

– суммарная хроматическая аберрация второго порядка.

2.5 Электронно-оптические свойства

2.5.1 Параксиальное приближение

Дифференциальные уравнения для пологих траекторий являются линейными однородными дифференциальными уравнениями второго порядка. Отсюда можно сделать заключение, что совокупность приосевых электронных траекторий, исходящих из перпендикулярного к оси элемента плоскости, создаёт электронно-оптическое изображение этой поверхности.

В этом можно убедиться следующим образом. Однородное линейное дифференциальное уравнение второго порядка имеет два независимых друг от друга решения. Общее его решение является линейной комбинацией этих двух частных решений. Это свойство характерно для линейных дифференциальных уравнений второго порядка. Исходя из этого свойства, докажем основной закон электронно-оптического изображения. Рассмотрим два частных интеграла $p(z)$ и $g(z)$ дифференциального уравнения (2.33), которые удовлетворяют в точке $z=z_0$ следующим условиям

$$\begin{cases} p(z_0) = 1, p'(z_0) = 0, \\ g(z_0) = 0, g'(z_0) = 1 \end{cases} \quad (2.40)$$

Эти решения не являются приосевыми траекториями, но любая приосевая траектория может быть получена из их линейной комбинации, так что

$$r(z) = ap(z) + bg(z), \quad (2.41)$$

где a и b – произвольные (соответственно малые) постоянные.

Соотношением

$$r(z_0) = ap(z_0) + bg(z_0) = a \quad (2.42)$$

Определена постоянная c_1 в соответствии с (2.40). Значение параметра c_2 определяется дифференцированием (2.41) и подстановкой $z=z_0$

$$r'(z_0) = ap'(z_0) + bg'(z_0) = b, \quad (2.43)$$

$$r(z) = r_0p(z) + r'_0g(z) \quad (2.44)$$

Параметр семейства $b = r'(z_0)$, который определяет начальное направление каждого луча, исходящего из P_0 пучка, обозначим как γ . Итак, пучок лучей, исходящих из P_0 электронов, определяется согласно (2.41), (2.42) и (2.43) уравнением

$$r(z) = r(z_0)p(z) + \gamma g(z) \quad (2.45)$$

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 (рисунок 2.2), координаты которых определяются уравнением

$$r(\bar{z}) = r(z_0)p(\bar{z}) + \gamma g(\bar{z}) \quad (2.46)$$

Если выбрать плоскость экрана так, чтобы при $z = z_1$ функция $g(z)$ имела следующую нулевую точку, то из (2.45) следует

$$r(z_1) = r(z_0)p(z_1) \quad (2.47)$$

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 $r_1=r_0p_1$. 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

$$M = p(z_1) \quad (2.48)$$

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 (2.48), а производная решения $g(z)$ – поперечное угловое увеличение

$$\Gamma = t'(z_1) \quad (2.49)$$

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 (2.40), имеем

$$z(F_0) = z_0 + \frac{g'(z_1)}{p'(z_1)}, \quad (2.50)$$

$$z(F) = z_1 - \frac{p(z_1)}{p'(z_1)}, \quad (2.51)$$

$$f_1 = \frac{1}{p'(z_1)}, \quad (2.52)$$

$$f_0 = -f_1 \sqrt{\frac{\Phi_0}{\Phi}} \quad (2.53)$$

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

2.5.2 Определение коэффициентов хроматических aberrаций

С учётом начальных условий, (2.39) можно записать в виде

$$\Delta r = \varepsilon(r_0 K_1 + r'_0 K_2)$$

где

$$K_1 = pJ_2 - gJ_1,$$

$$K_2 = pJ_4 - gJ_3$$

2.6 Используемые в работе численные методы

2.6.1 Метод Рунге-Кутты четвёртого порядка

Решение дифференциального уравнения второго порядка

$$y'' = f(x, y, y') \quad (2.54)$$

при начальных условиях

$$\begin{cases} y(x_0) = y_0, \\ y'(x_0) = y'_0 \end{cases} \quad (2.55)$$

методом Рунге-Кутты четвёртого порядка имеет следующий вид

$$y_{k+1} = y_k + y'_k \Delta x + \frac{1}{6}(k_1 + k_2 + k_3)\Delta x, \quad (2.56)$$

$$y'_{k+1} = y'_k + \frac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4) \quad (2.57)$$

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 относится к приёмам численного интегрирования. Получила название в честь британского математика Томаса Симпсона (1710—1761гг.).

Суть метода заключается в приближении подынтегральной 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 $[a, b]$

$$\int_a^b f(x)dx \approx \int_a^b p_2(x)dx = \frac{b-a}{6} \left(f(a) + 4f\left(\frac{a+b}{2}\right) + f(b) \right) \quad (2.58)$$

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 Руководство пользователя

Сразу после запуска программы «Исследование свойств пространственной фокусировки осесимметричной электронной линзы» появляется главная форма. На ней присутствуют такие элементы, как кнопки «Выход», «Справка», «Теория» и «Перейти к программе». Вверху формы расположено название программы. Окно главного меню представлено на рисунке 2.2.

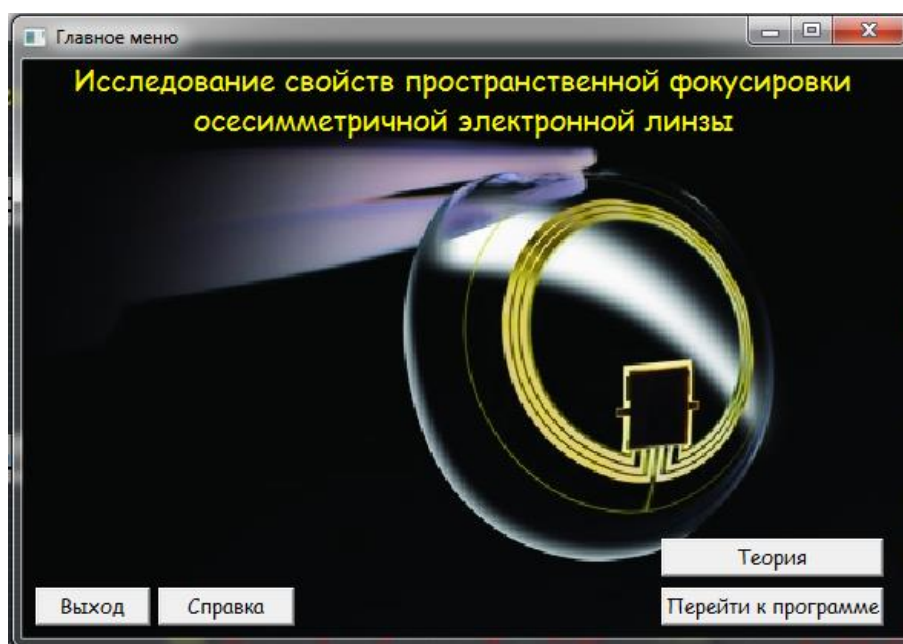


Рисунок 2.2 – Главное меню

При нажатии на кнопку «Выход» программа выйдет из режима эксплуатации, при нажатии на «Справка» программа перейдёт в окно, содержащее информацию о версии программы и об её разработчике.

Код, написанный на кнопки первой формы, представлен ниже.

```
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.

```

Окно справки представлено на рисунке 2.3.



Рисунок 2.3 – Справка о программе

При нажатии на кнопку «Назад» программа выйдет в главное меню.

При нажатии кнопки «Теория», находящейся в окне «Главное меню», откроется окно, содержащее теоретические факты об осесимметричных 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.

```

Окно теории представлено на рисунке 2.4.

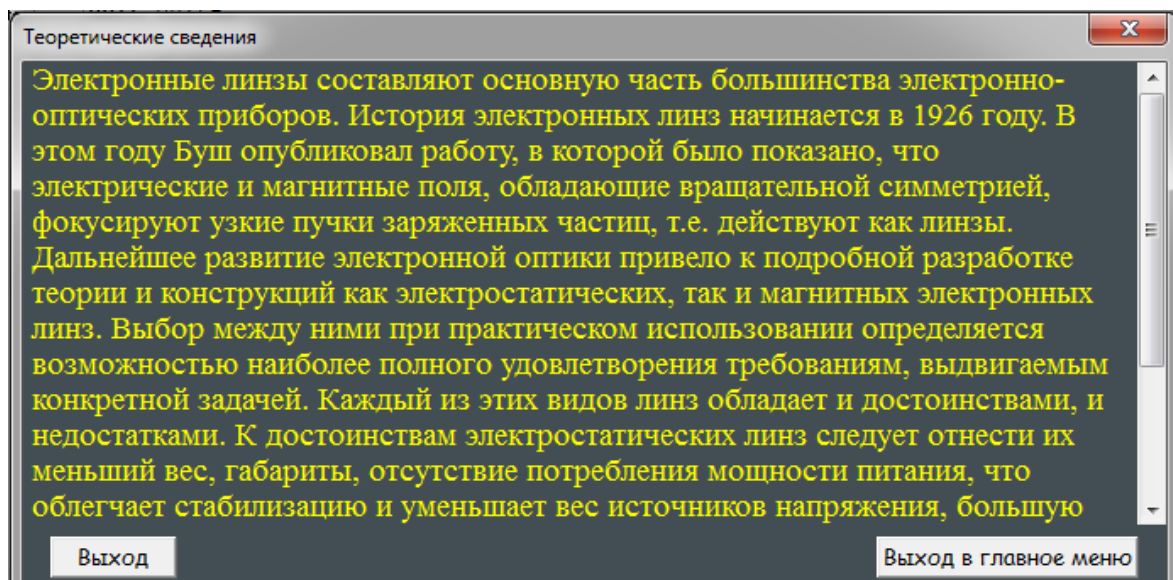


Рисунок 2.4 – Теоретические сведения

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 2.5.

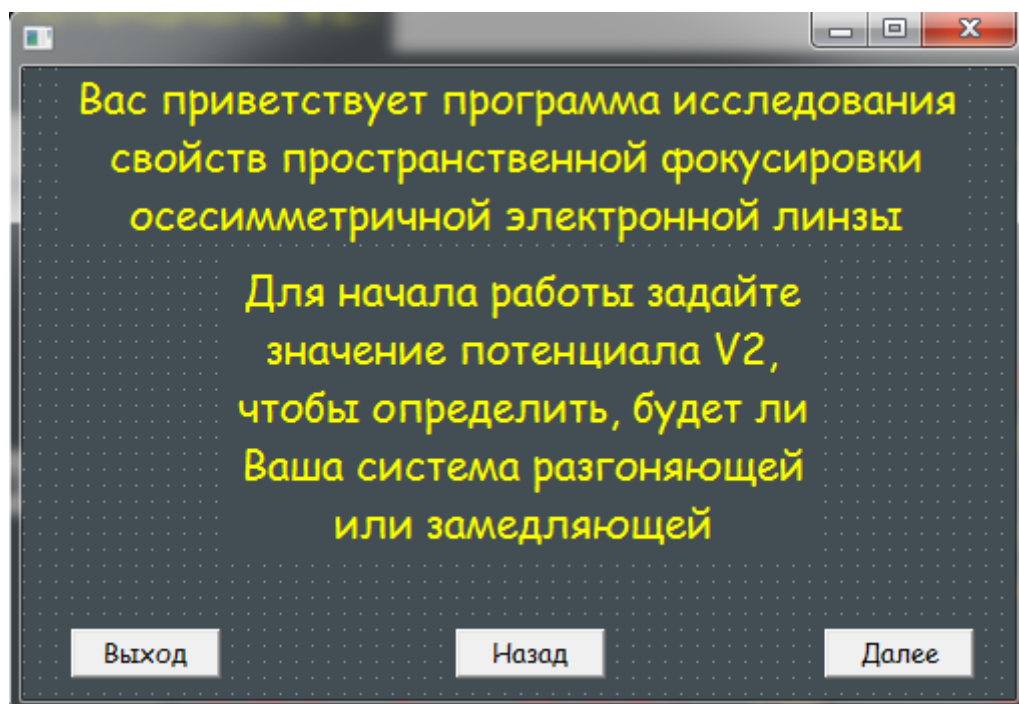


Рисунок 2.5 – Окно вступительной информации

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 2.6.

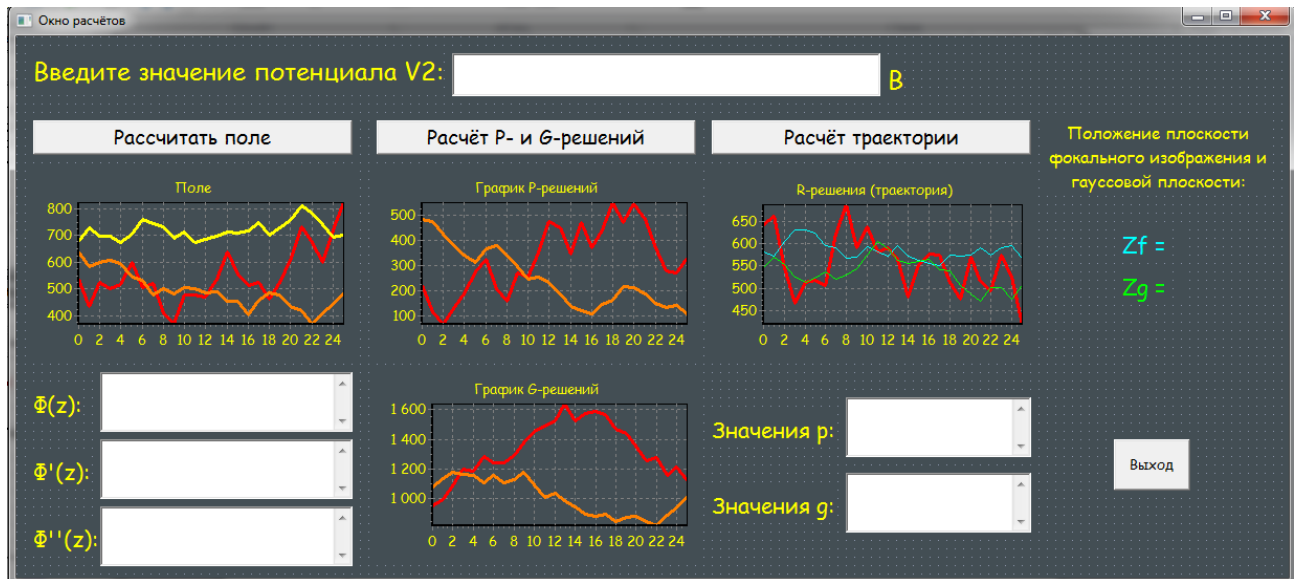


Рисунок 2.6 – Окно расчётов

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;

```

```

    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 «Расчитать 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 «Расчитать поле», программа произведёт расчёты, прописанные в отдельных процедурах. Такие же действия производятся и при нажатии на кнопки «Расчёт p- 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 .

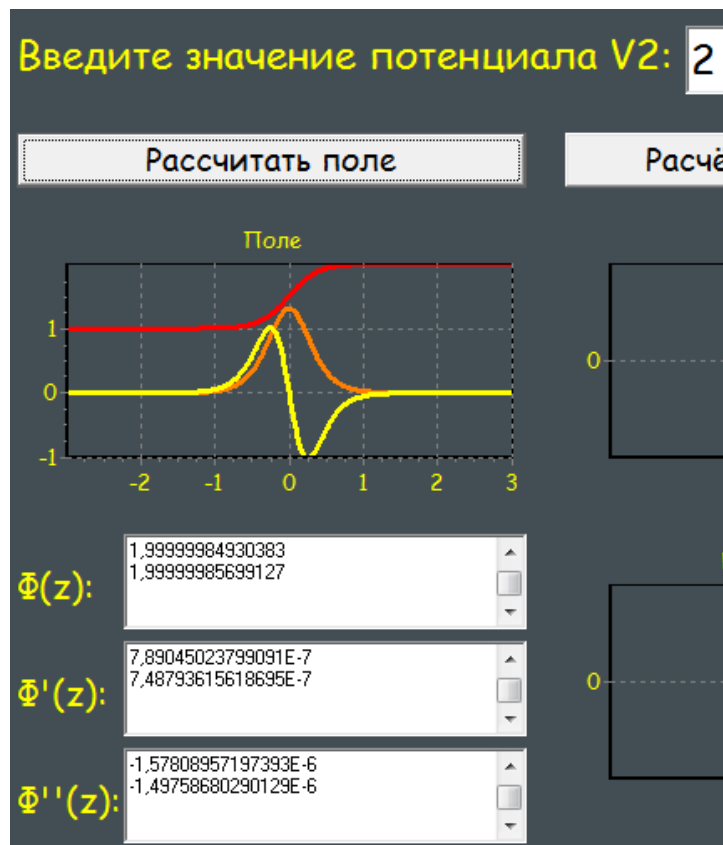


Рисунок 2.7 – Расчёт поля

На рисунке 2.8 представлен график расчета p- и g-решений.

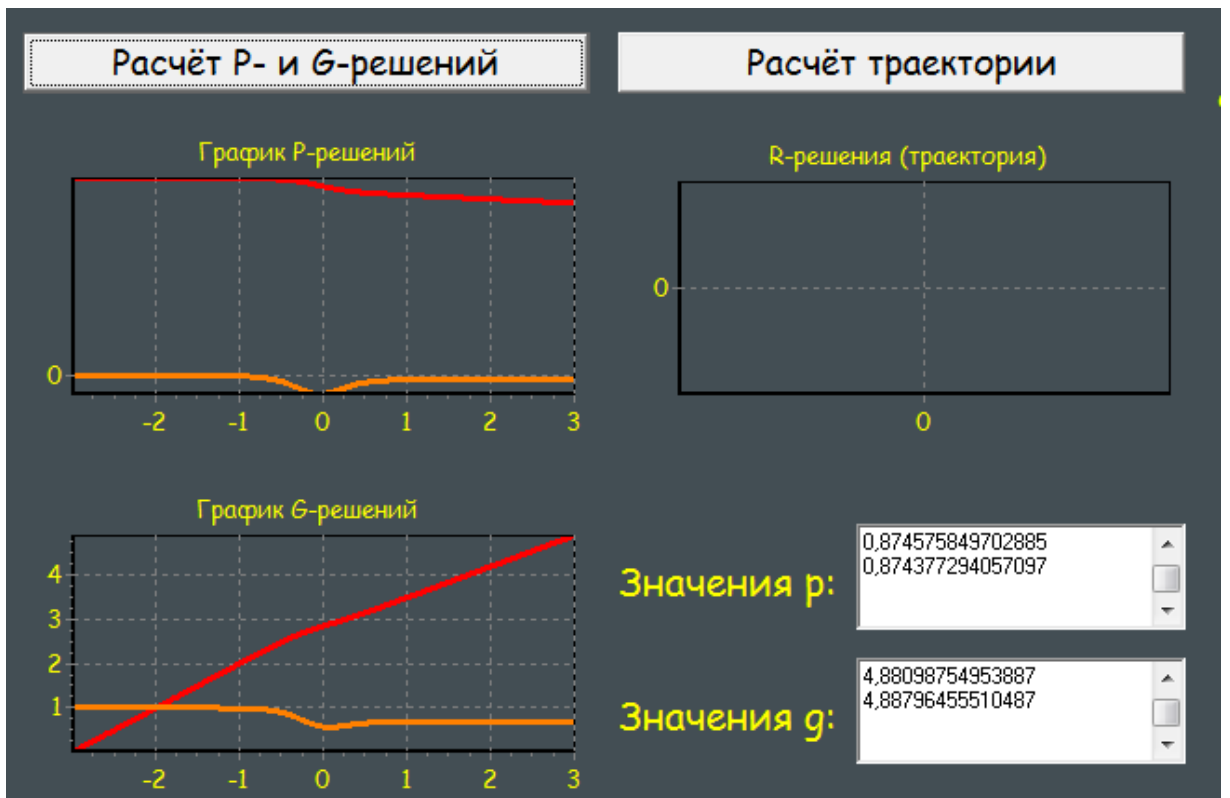


Рисунок 2.8 – Расчёт р- и g-решений

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.

При введении нового значения V_2 и нажатии по какой-либо из трёх кнопок построения графиков, вырисовывается новый график, основанный на введённом значении V_2 .

На рисунке 2.10 представлена общая картина работы программы в режиме эксплуатации.

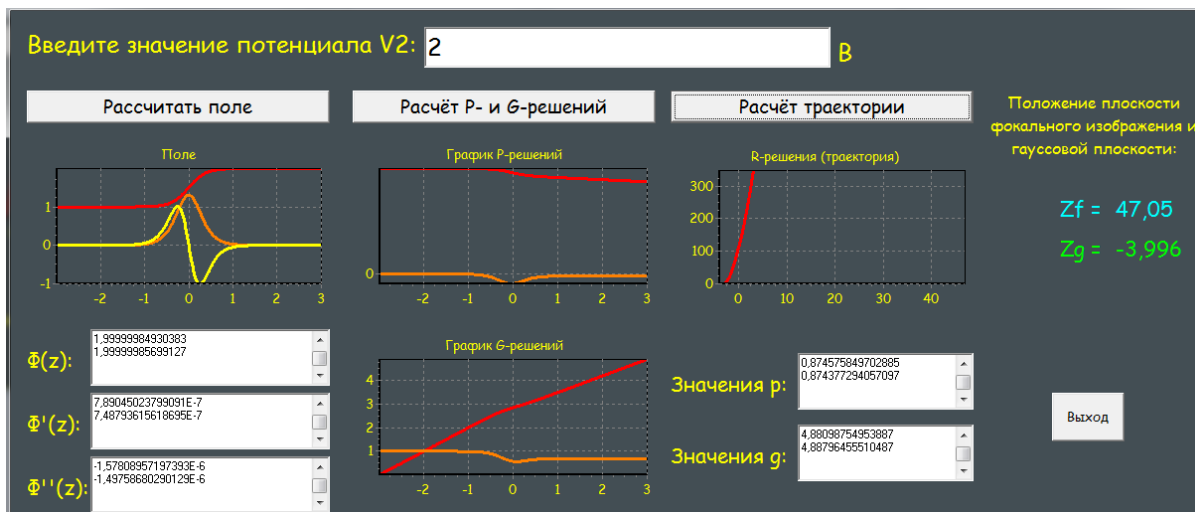


Рисунок 2.10 – Работа программы

Программный код, написанный на основную форму, где производятся все расчёты, представлен ниже.

```

unit Unit1;

interface

uses
  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;
  end;

```

```

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);
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 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;
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 := r1min;
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

Важнейшим фактором повышения эффективности производства в любой отрасли является улучшение управления. Качественный скачок на этом пути возможен только при использовании в управлении современных информационных технологий, так как управление – это непрерывный обмен информацией. Любой компьютер в процессе работы использует средства программного обеспечения.

Разработка программных средств требует определенных интеллектуальных и трудовых затрат, а также обязательного использования компьютерной техники, что определяет особенности расчета себестоимости программного продукта, которые учтены в данной методической разработке.

3.2 Трудовые ресурсы, используемые в работе

В работе были задействованы такие люди, как руководитель, занимавшийся постановкой задачи и разработкой основных бизнес-правил для работы проекта, инженер-разработчик, занимавшийся разработкой и отладкой программы для исследования фокусирующих свойств осесимметричной электронной линзы. Также были задействованы консультанты по экономической части и по части безопасности жизнедеятельности.

Общее количество участвующих в работе человек и их заработная плата представлены в таблице 3.1.

Т а б л и ц а 3.1 – Количество задействованных в проекте человек, и их заработная плата

Исполнитель	Количество, человек	Месячная заработная плата, тенге
Руководитель	1	60000
Консультант по части «Экономика»	1	30000
Консультант по части «Безопасность жизнедеятельности»	1	30000
Инженер-разработчик	1	80000
Итого	4	200000

3.3 Расчёт стоимости работы по проектированию и разработке

Разработка программного продукта – сложный и трудоемкий процесс, требующий финансовых затрат наряду с интеллектуальными и техническими. Любой процесс, начиная от работы компьютера и заканчивая вложенными человеческими усилиями, требует затраты определённых денежных средств. Производство расчёта стоимости разработки программного продукта является необходимой частью всего проекта. Затраты на разработку данного программного комплекса определяются по формуле

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

где ФОТ – фонд оплаты труда;

$C_{\text{пр}}$ – социальный налог;

A – амортизационные отчисления;

Э – затраты на электроэнергию;

$C_{\text{пр}}$ – прочие расходы;

H – накладные расходы.

Фонд оплаты труда рассчитывается по формуле

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

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

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

Труд сотрудников университета, задействованных в работе, оплачивается согласно положению о заработной плате АУЭС, труд программиста-разработчика принят условно, и равен 80000 тенге.

Основная заработная плата есть сумма заработных плат всех задействованных в работе лиц и рассчитывается по формуле

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

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

где x – инженер, руководитель, консультант;

T_x – время, затраченное на разработку ПП;

H_x – зарплата за один час.

Базовый показатель для определения составляющих затрат труда вычисляется по формуле

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

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

Программные продукты по степени новизны могут быть отнесены к одной из 4-х групп:

- 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 c определяется из таблицы 3.3, на пересечении ряда группы сложности и столбца степени новизны. Сложность разработки относится ко второй группе, в то время как степень новизны относится к группе А. Значит, коэффициент c равен 1,30.

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

Язык программирования	Группа сложности	Степень новизны				Коэф-т В
		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

Теперь, исходя из формулы (3.1), можно определить условное число команд Q

$$Q = 5000 * 1,30 = 6500$$

Далее необходимо рассчитать время, требующееся для создания программного продукта.

При разработке программного продукта было использовано техническое задание, согласно которому выполнение работ происходило последовательно по заданным пунктам, с учётом сроков их исполнения. График выполненных работ по разработке программного продукта представлен в таблице 3.4.

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	2	T _{ПЗ}
2	Описание задачи	3	T _{ОЗ}
3	Разработка алгоритма	5	T _А
4	Установка и запуск локального сервера	3	T _{ЛС}
5	Разработка модулей приложения	6	T _М
6	Разработка и создание интерфейса системы	5	T _{ИН}
7	Разработка основной части программы	6	T _{ОСН}
8	Тестирование модулей программы с целью выявления ошибок	2	T _{ТЕСТ}
9	Составление технической документации по программе	3	T _{ТЕХ}
10	Подготовка раздела экономической части	9	T _Э
11	Подготовка раздела безопасности жизнедеятельности	9	T _{БЖ}

Люди, задействованные в проекте, работают в различные промежутки времени, в течение которого реализуется проект, поэтому необходимо произвести расчёт дневной и почасовой оплаты труда.

Дневная заработная плата работника рассчитывается по формуле

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

где O – оклад работника в тенге;

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

– для руководителя

$$D = 60000 / 24 = 2500 \text{ тенге/день};$$

– для консультанта по экономической части

$$D = 30000 / 24 = 1250 \text{ тенге/день};$$

– для консультанта по части безопасности жизнедеятельности

$$D = 30000 / 24 = 1250 \text{ тенге/день};$$

– для инженера разработчика

$$D = 80000 / 24 = 3333,3 \text{ тенге/день}$$

Почасовая заработная плата работника, задействованного в проекте, рассчитывается по формуле

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

где D – заработная плата работника за один рабочий день,

z – количество часов рабочего дня (8 часов):

– для руководителя проекта

$$H = 2500 / 8 = 312,5 \text{ тенге/час};$$

– для консультанта по экономической части

$$H = 1250 / 8 = 156,25 \text{ тенге/час};$$

– для консультанта по части безопасности жизнедеятельности

$$H = 1250 / 8 = 156,25 \text{ тенге/час};$$

– для инженера-разработчика проекта

$$H = 3333,3 / 8 = 416,6 \text{ тенге/час}$$

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

Определим время, затраченное на каждый этап создания программного продукта в отдельности

1) $T_{пз}$ (время на подготовку описания задачи) берётся по факту и составляет (от 3-х до 5-ти дней по 8 часов): 24 чел/час;

2) $T_{оз}$ (время на описание задачи) определяется по формуле

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

где В – коэффициент учёта изменений задачи. Коэффициент В, в зависимости от сложности задачи и числа изменений, выбирается в интервале от 1,2 до 1,5 (таблица 3.3), равный 1,35; К – коэффициент, учитывающий квалификацию программиста.

Выберем 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), посчитаем время на описание задачи

$$T_{\text{оз}} = 6500 * 1,35 / (50 * 1) = 175,5 \text{ чел/час}$$

3) T_a (время на разработку алгоритма) рассчитывается по формуле

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

Тогда, применяя формулу (3.9), получаем

$$T_a = 6500 / (50 * 1) = 130 \text{ чел/час};$$

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

$$T_{\text{лс}} = 130 \text{ чел/час};$$

5) $T_{\text{бс}}$ (время на разработку блок-схемы) определяется аналогично T_a по формуле (3.9) и составляет 130 чел/час;

6) T_m (время на разработку модулей программы) определяется по формуле

$$T_m = Q * 1,5 / (50 * K) \quad (3.10)$$

Применяя формулу (3.10), подсчитываем T_m

$$T_m = 6500 * 1,5 / (50 * 1) = 195 \text{ чел/час};$$

7) $T_{ин}$ (время на отладку и тестирование программы) определяется аналогично T_m

$$T_{ин} = 6500 * 1,5 / (50 * 1) = 195 \text{ чел/час};$$

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) $T_{осн}$ (время на разработку основной части программы) определяется по формуле

$$T_{осн} = 6500 * 1,5 / (50 * 1) = 453,66 \text{ чел/час};$$

10) $T_{тест}$ (время на тестирование программы), определяется по формуле

$$T_{тест} = 6500 * 1,5 / (50 * 1) = 303,6 \text{ чел/час};$$

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

$$T_{тех} = 24 \text{ чел/час};$$

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

$$T_э = T_{бж} = 50 \text{ чел/час};$$

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

$$T_p = 80 \text{ чел/час},$$

$$T_{эк} = T_{бжд} = 22 \text{ чел/час}$$

Трудоёмкость разработки программного продукта рассчитывается по формуле

$$T_{ст} = T_{пз} + T_{оз} + T_A + T_{лс} + T_{бд} + T_m + T_{ин} + T_{осн} + T_{тест} + T_{тех} + T_э + T_{бж} \quad (3.11)$$

$$T_{ст} = 24 + 175,5 + 130 + 130 + 130 + 195 + 195 + 24 + 453,66 + 303,6 + 32 + 50 + 50 = 1892,76 \text{ чел/час}$$

3.4 Расчёт затрат на разработку программного обеспечения

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

$$З_{\text{осн}} = 1892,76 * (312,5 + 156,25 + 156,25 + 416,6) = 1971498,8 \text{ тг}$$

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

$$З_{\text{доп}} = З_{\text{осн}} * 10\% \quad (3.12)$$

$$З_{\text{доп}} = 1971498,8 * 0,1 = 197149,88 \text{ тг}$$

Общий фонд оплаты труда согласно формуле (3.2) составит

$$\text{ФОТ} = 1971498,8 + 197149,88 = 2168648,68 \text{ тг}$$

Социальный налог составляет 11 процентов от ФОТ и рассчитывается по формуле

$$С_{\text{н}} = (\text{ФОТ} - \text{ПО}) * 11\% \quad (3.13)$$

ПО (пенсионные отчисления) составляют 10 процентов от ФОТ и рассчитываются по формуле

$$\text{ПО} = \text{ФОТ} * 10\% \quad (3.14)$$

Размер пенсионных отчислений согласно формуле (3.14) составляет

$$\text{ПО} = 2168648,68 * 0,1 = 216864,87 \text{ тг}$$

Отчисления по социальному налогу вычисляются по формуле (3.13) и составляют

$$С_{\text{н}} = (2168648,68 - 216864,87) * 0,11 = 214696,22 \text{ тг}$$

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

$$A = \frac{\text{Собор} * \text{На} * N}{100 * 12 * t} \quad (3.15)$$

где H_A – норма амортизации;

$C_{\text{обор}} = 395500$ тг – первоначальная стоимость оборудования;

N – количество дней на выполнение работ;

t – общее время использования персонального компьютера.

В таблице 3.6 представлена стоимость оборудования, использованного при создании проекта.

Т а б л и ц а 3.6 – Стоимость оборудования

Наименование	Модель	Стоимость
Процессор	3.5GHz quad-core Intel Core i7 (Turbo Boost up to 3.9GHz).	200 000
Монитор	27-inch (diagonal) LED-backlit display with IPS technology; 2560-by-1440 resolution with support for millions of colors	120 500
Клавиатура	Apple Wireless Keyboard Full-size keyboard with 78 (U.S.) or 79 (ISO) keys, including 12 function keys and 4 arrow keys (inverted “T” arrangement)	20 000
Компьютерная мышь	Magic Mouse Smooth, seamless Multi-Touch surface with support for simple gestures (scrolling and swiping).	10 000
Модем	Apple Airport Time Capsule	45 000
Итого		395 500 тг

Норма амортизации (H_A), рассчитывается по формуле

$$H_A = ((C_{\text{обор}} - C_{\text{ликв}}) / (T_{\text{норм}} * C_{\text{обор}})) * 100\% \quad (3.16)$$

$$H_A = ((395500 - 22405,075) / (4 * 395500)) * 100 = 23,58\%$$

где $C_{\text{ликв}} = 0,05665 * 395500 = 22405,075$ тг – ликвидационная стоимость;

$T_{\text{норм}}$ – нормативный срок службы (для ПК – 4 года).

Общее время использования персонального компьютера рассчитывается по формуле

$$t = T_{\text{лс}} + T_{\text{бд}} + T_{\text{м}} + T_{\text{ин}} + T_{\text{осн}} + T_{\text{тест}}, \quad (3.17)$$

$$t = 76,67 + 76,67 + 115 + 115 + 302,7 = 705,34 \text{ ч}$$

Отсюда

$$A = (395500 * 23,58 * 4,57)/(100 * 12 * 29,39) = 1208,37 \text{ тг}$$

Затраты на электроэнергию вычисляются по формуле

$$C_{\text{ээ}} = M * k_3 * t * C_{\text{кВт/ч}}, \text{ тг}, \quad (3.18)$$

$$C_{\text{ээ}} = 0,45 * 0,8 * 705,34 * 13,39 = 3400,02 \text{ тг}$$

где M – мощность ЭВМ ($450 \text{ Вт} = 0,45 \text{ кВт}$);

k_3 – коэффициент загрузки ($0,8$);

$C_{\text{кВт/ч}} = 13,39 \text{ тг}$ – стоимость 1 кВт-час электроэнергии;

t – время работы, час.

Расходы на материалы и комплектующие, используемые для написания программного продукта ($C_{\text{мик}}$), а также затраты на техническое обслуживание и ремонт ($C_{\text{то}}$), вычисляются по формулам

$$C_{\text{мик}} = 0,01339 * C_{\text{обор}}, \text{ тг}, \quad (3.19)$$

$$C_{\text{мик}} = 0,01339 * 395500 = 5295,745 \text{ тг},$$

$$C_{\text{то}} = 0,02781 * C_{\text{обор}}, \text{ тг}, \quad (3.20)$$

$$C_{\text{то}} = 0,02781 * 395500 = 10998,855 \text{ тг}$$

Накладные расходы, связанные с управлением и обслуживанием оборудования и прочими дополнительными затратами на обеспечение процессов производства и обращения, вычисляются по формуле

$$C_{\text{накл}} = 0,618 * \text{ФОТ}, \text{ тг}, \quad (3.21)$$

$$C_{\text{накл}} = 0,618 * 2168648,68 = 1340225 \text{ тг}$$

Сводные результаты расчёта себестоимости программного продукта предоставлены таблице 3.7.

Т а б л и ц а 3.7 – Себестоимость разработки программного продукта

Статья затрат	Сумма, тенге	Доля от общей ст-ти, %
Фонд оплаты труда	2168648,68	58,17
Затраты на электроэнергию	3400,02	0,09
Социальный налог	214696,22	5,75
Накладные расходы	1340225	35,95
Амортизационные отчисления	1208,37	0,003
Итого	3728178,29	100

3.5 Цена интеллектуального труда

Чистая прибыль от разработки сайта рассчитывается по формуле

$$\text{ЧП} = \text{С} * 15\%, \quad (3.19)$$

$$\text{ЧП} = 3728178,29 * 0,15 = 559226,74 \text{ тг},$$

$$\text{Р} = \text{С} + \text{ЧП} \quad (3.20)$$

где Р – цена реализации продукта, и равна

$$\text{Р} = 3728178,29 + 559226,74 = 4287405,03 \text{ тг}$$

Тогда, НДС от программного продукта составит

$$\text{НДС} = \text{Р} * 12\%, \quad (3.21)$$

$$\text{НДС} = 4287405,03 * 0,12 = 514488,6 \text{ тг}$$

Сводные результаты расчёта себестоимости разработки ПО и ее структура представлены в таблице 3.7 и на рисунке 3.1.

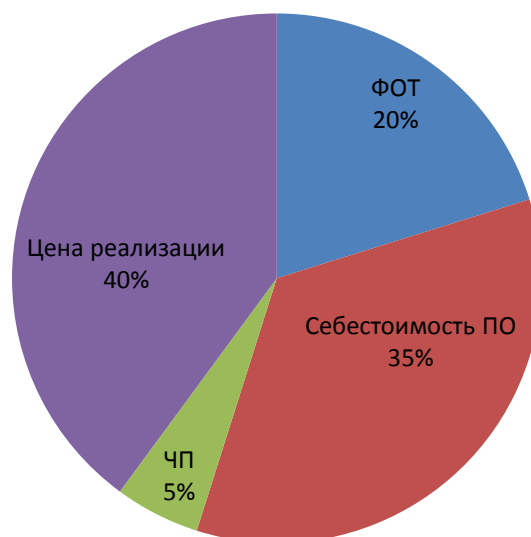


Рисунок 3.1 – Структура себестоимости разработки программного продукта «Исследование свойств пространственной фокусировки осесимметричной электронной линзы»

3.6 Вывод

Себестоимость разработки программного продукта составила 3728178,29 тенге, наибольшую долю в разработке составил фонд оплаты труда (58,17%). Цена реализации составила 4287405,03 тенге, в то время как цена аналогов варьируется в пределах от 2 миллионов до 7 миллионов тенге. Данный программный продукт является рентабельным, актуальным и эффективным.

4 Безопасность жизнедеятельности и труда человека

4.1 Анализ вредных факторов, влияющих на здоровье программиста

Вопросы, касающиеся безопасности жизни человека, нужно решать на всех стадиях жизненного цикла, независимо от того, какой это цикл – разработка программы, внедрение её в жизнь, её тестирование и эксплуатация. Обеспечение безопасности деятельности, прежде всего, зависит от правильной оценки опасных факторов, влияющих на человека в процессе разработки. Негативные изменения в организме человека могут быть вызваны рядом причин – это могут быть как факторы производственной среды, чрезмерная эмоциональная или физическая нагрузка, так и разное сочетание этих факторов.

В данной части дипломного проекта рассматриваются и решаются вопросы безопасности жизнедеятельности на стадии разработки программного продукта, предназначенного для исследования свойств пространственной фокусировки осесимметричных электронных линз, а также вопросы анализа опасных и вредных факторов, воздействующих на программиста при разработке данного программного продукта.

Опасные и вредные производственные факторы по природе возникновения делятся на такие группы, как химические, физические, биологические и психофизиологические.

В рабочем помещении на человека могут негативно повлиять следующие физические факторы:

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

К химически опасным факторам, постоянно действующим на программиста при разработке программного продукта, относится возникновение активных частиц в результате ионизации воздуха при работе компьютера.

Биологические вредные производственные факторы в данном помещении не наблюдаются.

К психологически вредным факторам, воздействующим на программиста в течение его рабочего времени, можно отнести следующие:

- нервно-эмоциональные перегрузки;

- умственное напряжение;
- перенапряжение зрительного анализатора.

Далее более подробно рассмотрены опасные и вредные факторы, воздействующие на программиста, возникшие при разработке данного программного продукта.

4.2 Микроклимат рабочей зоны программиста

Микроклимат производственных помещений – это климат внутренней среды этих помещений, который определяется действующими на организм человека сочетаниями температуры, влажности и скорости движения воздуха.

Рабочее помещение является помещением, где выполняется основная часть работы над проектом, а также лёгкие физические работы, поэтому должны соблюдаться следующие требования: оптимальная температура воздуха – 22°C (допустимая – 20-24°C), оптимальная относительная влажность – 40-60% (допустимая – не более 75%), скорость движения воздуха не должна превышать 0,1м/с.

Для создания и поддержания в рабочем помещении независимо от наружных условий, в холодное время года используется водяное отопление, в тёплое время года применяется кондиционирование воздуха. Кондиционер представляет собой вентиляционную установку, которая с помощью внутренних устройств регулирования поддерживает в помещении заданную температуру и влажность окружающего пространства. В таблице 4.1 представлены параметры микроклимата для помещений в холодный и тёплый период года.

Т а б л и ц а 4.1 – Параметры микроклимата для помещений

Период года	Параметр микроклимата	Величина
Холодный	Температура воздуха в помещении	22 – 24°C
	Относительная влажность	40 – 60%
	Скорость движения воздуха	до 0,1м/с
Теплый	Температура воздуха в помещении	23 – 25°C
	Относительная влажность	40 – 60%
	Скорость движения воздуха	0,1 – 0,2м/с

Объём помещения, в котором размещены работники и программисты, не должен быть меньше, чем 19,5м³/чел с учётом максимального числа одновременно работающих в одном помещении человек. Нормы подачи свежего воздуха в рабочем помещении, где расположены компьютеры, приведены в таблице 4.2.

Т а б л и ц а 4.2 – Нормы подачи свежего воздуха

Характеристика помещения	Объемный расход подаваемого в помещение свежего воздуха, м ³ / на одного человека в час
Объём до 20м ³ на человека	Не менее 30
20 – 40м ³ на человека	Не менее 20
Более 40м ³ на человека	Естественная вентиляция

4.3 Освещение рабочего места

Работа программиста происходит в помещении, поэтому освещение в нём должно соответствовать оптимальным условиям зрительного солнечного освещения во избежание травм. Требования к освещённости к помещениям, где установлены компьютеры, следующие: при выполнении зрительных работ высокой точности общая освещённость должна составлять 300 лк, а комбинированная – 750 лк. Аналогичные требования при выполнении работ средней точности – 200 и 300 лк соответственно. При этом монитор и источники света должны быть расположены под таким углом друг к другу, чтобы не создавать бликов на поверхности экрана и не мешать работе. Кроме того всё пространство, находящееся в поле зрения программиста, должно быть освещено достаточно равномерно – это основное гигиеническое требование к освещению. Другими словами, степень освещения помещения и яркость экрана компьютера должны примерно совпадать, так как яркий свет в районе бокового зрения значительно увеличивает напряжённость глаз и приводит к их быстрой утомляемости.

Помещение, в котором находятся вычислительные машины, должно иметь как естественное, так и искусственное освещение. При этом искусственное освещение может применяться как в тёмное, так и в светлое время суток. Естественное освещение должно осуществляться через светопрёмы, которые в большинстве своём должны быть ориентированы на север и северо-восток, и обеспечивать коэффициенты естественной освещённости не ниже 1,2 процента в зонах с постоянными холодами и 1,5 процента на остальных территориях. Искусственное освещение в помещениях эксплуатации компьютеров должно осуществляться системой общего равномерного освещения. В качестве источников искусственного освещения можно использовать люминесцентные лампы, которые можно попарно объединить в светильники, равномерно располагающиеся над рабочими поверхностями.

При таком расположении компьютеров, при котором они стоят в ряд, следует выполнять общее в виде сплошных или прерывистых линий светильников, расположенных сбоку от рабочих мест, параллельно линии зрения пользователя. При расположении компьютеров по периметру, линии

светильников должны располагаться прямо над рабочим столом, ближе к его переднему краю, обращённому к программисту.

Если правильно выбрать светильники и расположить рабочие места в правильном положении относительно естественных и искусственных источников освещения, то за счёт этого можно уменьшить отражённую блеклость на рабочих поверхностях, при этом яркость бликов на экране не должна превышать 40 кд/кв.м, а яркость потока при изменении системы отражённого освещения не должна превышать 200 кд/кв.м.

Для того, чтобы исключить блики, появляющиеся на экране монитора от светильников, необходимо использовать антибликовые, или, как их часто называют, антибликсерные сетки, специальные фильтры для экранов, или располагать источники света с обеих сторон параллельно направлению взгляда на экран.

Поскольку экран монитора тоже является источником света, при постоянном чтении информации с него происходит быстрое утомление глаз, особенно если яркость свечения монитора установлена слишком высокой. Также раздражение глаз вызывает мерцание изображения на мониторе, вызванной низкой частотой кадровой развёртки. Рекомендуемая частота кадров развёртки для уменьшения мерцания монитора – не ниже 75 Гц для ЭЛТ-мониторов, для ЖКИ-мониторов достаточной является минимальная частота кадров в 60 Гц.

4.4 Воздействие электромагнитных излучений и электрического поля. Статическое электричество

Монитор, помимо того, что является источником освещения и тепла, также является и сильным источником электромагнитного излучения. Особенно много излучения приходится на его боковые и задние стенки, так как они не имеют специального защитного покрытия, которое есть у лицевой части экрана. Наибольшее влияние, которое электромагнитное излучение оказывает на человека, оно оказывает на нервную, иммунную и половую систему.

Действие электромагнитного излучения компьютера на иммунную систему проявляется в следующем: иммунная система уменьшает выброс в кровь ферментов, выполняющих защитную функцию организма, происходит ослабление системы клеточного иммунитета. Эндокринная система, в свою очередь, начинает выбрасывать в кровь большее количество адреналина, чем требует организм, и, как следствие, возрастает нагрузка на сердечнососудистую систему организма человека. Происходит сгущение крови, в результате чего клеткам не достаёт кислорода. В качестве защитных мер от электромагнитных излучений компьютера, нужно совершать регулярные прогулки на свежем воздухе, проводить регулярное проветривание помещения, заниматься спортом, соблюдать элементарные правила работы, проводить работу с хорошей техникой, которая удовлетворяет всем стандартам безопасности и санитарным нормам.

Последствия регулярной работы с компьютером без применения защитных средств могут повлечь за собой заболевания органов зрения, возникающие у подавляющего большинства пользователей, болезни сердечнососудистой системы, всевозможные кожные заболевания, опухоли, болезни кишечника и другие.

Защита от электромагнитного излучения компьютера состоит из следующих пунктов:

1) по возможности, стоит приобрести жидкокристаллический монитор, так как его электромагнитное излучение намного меньше, чем у распространённых мониторов, состоящих из электронно-лучевых трубок;

2) системный блок и монитор должны находиться как можно дальше от программиста, поскольку тоже обладают электромагнитным излучением;

3) нельзя оставлять компьютер включённым на длительное время если он не используется. Это ускорит износ компьютера и может негативно сказаться на здоровье. Также, рекомендуется использовать «спящий режим» для монитора – это уменьшит время разрядки компьютера и ограничит программиста от воздействия вредных электромагнитных лучей;

4) в связи с тем, что электромагнитное излучение от стенок монитора намного больше, постарайтесь поставить монитор в угол, так что бы излучение поглощалось стенами. Особое внимание стоит обратить на расстановку мониторов в офисах;

5) по возможности необходимо сократите время работы за компьютером и почаще прерывайте работу;

б) компьютер должен быть заземлён. Если был приобретён защитный экран, то его тоже следует заземлить, для этого специально предусмотрен провод, на конце которого находится металлическая прищепка (которую не следует цеплять к системному блоку).

4.5 Шум и вибрация

Шум является совокупностью звуков различной частоты, интенсивности и продолжительности. Высокий уровень шума, создаваемый всевозможными машинами, является неотъемлемой частью нынешнего мира. Шум от печатных устройств, множительной техники, оборудования для кондиционирования воздуха, вентиляторов систем охлаждения и трансформаторов в компьютерах хоть и не является смертельным, но точно является одним из неблагоприятных факторов производственной среды программиста.

Длительное действие шума высокой интенсивности (свыше 80 децибел) может привести к патологиям слуховых органов и негативно повлиять на нервную систему человека. Шум может также привести к быстрой утомляемости человека, что в свою очередь ведёт к производственным ошибкам, связанным со всевозможными расчётами.

Для снижения уровня шума следует выполнять шумоизоляцию стен и потолка помещения, где установлены компьютеры. Стены также могут быть

облицованы звукопоглощающими материалами с применением различных звукопоглощающих устройств. Шумящее оборудование, уровни шума которого превышают нормированные, должно находиться вне помещения.

К средствам индивидуальной защиты можно отнести противошумные наушники, вкладыши, шлемы и каски, специальные костюмы, беруши.

Вибрация – это колебания точки или механической системы под воздействием каких-либо возбудителей. Вибрация может вызвать профессиональные заболевания, при которых изменяются физиологические и психические функции организма человека. Уровень вибрации в помещениях вычислительных центров может быть снижен путём установки противошумного или звукоизолирующего оборудования на рабочих местах.

Правильная планировка помещения и организация рабочего места, а также правильное размещение оборудования являются очень важными факторами, позволяющими снизить шум и вибрацию.

4.6 Другие вредные факторы, оказывающие воздействие на здоровье программиста

Работа программиста – сидячая и малоподвижная. Следствиями малоподвижной работы могут стать излишний вес, сердечнососудистые заболевания, повышенное артериальное давление, атеросклероз, затекание мышц, спазмы в мышцах и другие.

Малоподвижность приводит к заболеваниям опорно-двигательного аппарата, таким как остеохондроз, радикулит, артрит. Работа за компьютером ведёт к возникновению геморроя и затеканию мышц и частей тела вследствие непоступления достаточного количества крови в органы. В целях профилактики этих болезней необходимо организовывать мероприятия активного отдыха, занятия спортом, пешие прогулки на свежем воздухе. Также необходимы определённые разминки в перерывах при работе над проектом.

Постоянная работа с клавиатурой может привести к хронической усталости в пальцах, а также к серьёзным заболеваниям суставов кистей рук. Лёгкая разминка пальцев и их расслабление значительно снизят ущерб суставам.

Работа программиста является источником нервных и психических заболеваний. Стрессовыми ситуациями, которые расшатывают нервную систему, являются такие ситуации, как потеря важной информации, сбой в работе компьютера. Системное программирование, которое требует постоянных математических расчётов, подчас приводит к психическим заболеваниям, лечение которых осуществляется уже принудительными способами и методами.

Из всего вышесказанного можно сделать вывод о том, что различные производственных факторы в основном негативно влияют на здоровье человека и могут вызвать профессиональные заболевания различной степени тяжести.

4.7 Характеристики рабочего помещения

В данном дипломном проекте рассматривается рабочее помещение, расположенное в здании, которое не находится в непосредственной близости от железнодорожной магистрали или аэропорта, поэтому можно считать, что внешних источников шума, влияющих на процесс работы, нет. Ниже описываются характеристики выбранного помещения:

- 1) рабочее помещение находится на третьем этаже пятиэтажного здания;
- 2) размеры рабочего помещения (комнаты): длина 6 метров, ширина 3 метра, высота 3 метра;
- 3) остекление помещения – двойное (одно окно размером 2400x2000 мм);
- 4) искусственное освещение – светильники: два светильника, в каждом по две лампы, являющиеся люминесцентными, с возможностью объединения двух ламп в одну;
- 5) внутренняя отделка стен – светлая;
- 6) помещение по зрительным условиям работы относится к четвёртому разряду (размер различаемых при работе предметов от 1 до 10 мм и выше);
- 7) установлен один оконный кондиционер Samsung.

4.8 Используемое оборудование

В разработке дипломного проекта был задействован один персональный переносной ноутбук. Технические характеристики:

- 1) процессор Intel Core i3, 2,4 GHz, 2 MB RAM / HDD 240 Gb;
- 2) монитор Samsung 943n, диагональ 19”;
- 3) габариты: 1200x750x1150 (персональный компьютер + стол);
- 4) электропитание: переменное напряжение 220-250 В, с частотой 50 Гц и мощностью 400 Вт.

Осветительное оборудование, использованное при разработке проекта:

- 1) люминесцентные лампы, 2 светильника по 2 лампы;
- 2) электропитание: переменное напряжение 220-250 В, с частотой 50 Гц, мощность каждого светильника составляет 65 Вт.

Электротехническое оборудование является потенциальным источником возникновения пожарной опасности, поэтому ниже будет произведён расчёт противопожарной безопасности.

4.9 Расчётная часть

4.9.1 Инженерный расчёт электромагнитного излучения от компьютера

Исходные данные для расчёта электромагнитного поля от монитора:

- $C = 3 \cdot 10^8$ м/с – скорость распространения волны;

- $L = 0,28 \text{ мм} = 2,8 \cdot 10^{-4}$ – длина диполя;
- $I = 0,15 \text{ А}$ – амплитуда тока;
- $E_0 = 8,85 \cdot 10^{-12}$ – электрическая постоянная;
- $f = 100 \text{ Гц}$ – частота работы системы кадровой развертки монитора;
- $t = 30 \text{ мин} = 1800 \text{ с}$ – время непрерывной работы за монитором;
- $r = 0,5 \text{ м}$ – расстояние от центра монитора до точки наблюдения;
- $\nu = 10 \text{ град} = 0,175 \text{ рад}$ – угол между нормалью к центру экрана и направлением на точку наблюдения;
- $\omega = 2 \cdot \pi \cdot f = 628,319 \text{ рад/с}$ – круговая частота электромагнитного поля, создаваемого системой кадровой развертки.

Расчёт:

Определим длину волны кадровой развёртки монитора, излучаемой системой

$$\lambda = c / f = 3 \cdot 10^8 / 100 = 3 \cdot 10^6 \text{ м}$$

Так как расстояние от центра монитора до точки наблюдения намного меньше полученной длины волны, то расчёт напряжённости электромагнитного поля будет произведён по формулам

$$E_r = (I_0 \cdot L \cdot \cos(\nu) \cdot \sin(\omega \cdot t)) / 2 \cdot \pi \cdot \omega \cdot \epsilon_0 \cdot r_3, \text{ В/м}, \quad (4.1)$$

$$E_\nu = I_0 \cdot L \cdot \sin(\nu) \cdot \sin(\omega \cdot t) / 4 \cdot \pi \cdot \omega \cdot \epsilon_0 \cdot r_3, \text{ В/м}, \quad (4.2)$$

$$H_\alpha = I_0 \cdot L \cdot \sin(\nu) \cdot \cos(\omega \cdot t) / 4 \cdot \pi \cdot r_2, \text{ А / м}, \quad (4.3)$$

$$E = \sqrt{(E_{r2} + E_{\nu2})}, \text{ В/м} \quad (4.4)$$

где E , H – напряжённость электрического и магнитного поля в точке наблюдения. Подставив исходные данные в формулы (4.1) и (4.4), получим напряжённость поля, создаваемую одним пикселем монитора

$$E_r = (0,15 \text{ А} \cdot 0,00028 \text{ м} \cdot \cos(0,175 \text{ рад}) \cdot \sin\left(\frac{628,319 \text{ рад}}{c}\right) \cdot 1800 \text{ с}) / (2 \cdot \pi \cdot 628,319 \cdot 8,85 \cdot 10^{-12} \cdot 0,5^3) = 3,86 \cdot 10^{-7} \text{ В/м},$$

$$E_\nu = (0,15 \text{ А} \cdot 0,00028 \text{ м} \cdot \cos(0,175 \text{ рад}) \cdot \sin\left(\frac{628,319 \text{ рад} \cdot 1800}{c}\right) \cdot 1800 \text{ с}) / (4 \cdot \pi \cdot 628,319 \cdot 8,85 \cdot 10^{-12} \cdot 0,5^3) = 3,40 \cdot 10^{-8} \text{ В/м},$$

$$H_\alpha = (0,15 \text{ А} \cdot 0,00028 \text{ м} \cdot \sin(0,175 \text{ рад}) \cdot \cos\left(\frac{628,319 \text{ рад} \cdot 1800}{c}\right) \cdot 1800 \text{ с}) / (4 \cdot \pi \cdot 0,5^2) = 2,07 \cdot 10^{-6} \text{ В/м},$$

$$E = \sqrt{((3,86 \cdot 10^{-7})^2 + (3,40 \cdot 10^{-8})^2)} = 2,07 \cdot 10^{-6} \text{ В/м}$$

Для простоты расчётов допустим, что напряжённости, создаваемые каждым пикселем монитора, одинаковы. Тогда напряженность поля, создаваемая N ($N = 9,6 \cdot 10^5$) точками (т.е. экран размером 1200 x 800 пикселей), будет рассчитываться по формулам (4.5) и (4.6)

$$E_{\Sigma} = 10 \lg (N \cdot 10^{E_{\Sigma} N}), \text{ В/м}, \quad (4.5)$$

$$H_{\Sigma} = 10 \lg (N \cdot 10^{H_{\Sigma} N}), \text{ А/м} \quad (4.6)$$

После подстановки N , E_{Σ} , H_{Σ} в формулы (4.5) и (4.6), получим

$$E_{\Sigma} = 6,645 \text{ В/м},$$

$$H_{\Sigma} = 0,652 \text{ А/м}$$

Приведённые выше расчёты показывают, что использованные при разработке проекта монитора соответствуют нормам.

Чтобы уменьшить влияние вредных факторов, влияющих на человека, необходимо расположить экран дисплея немного выше уровня глаз. Это создаст разгрузку тех групп окологлазных мышц, которые наиболее напряжены при обычном направлении взгляда – вниз или вперёд.

4.9.2 Расчёт противопожарной безопасности

Рассматриваемое помещение, используемое для разработки проекта, по взрывопожарной и пожарной опасности соответствует категории В (пожароопасное помещение) в соответствии со СНиП РК 3.02-04-2009 «Административные и бытовые здания» в разделе «Противопожарные требования», так как в нём имеются горючие материалы (мебель, бумага, покрытие пола, шторы, пыль), которые горят при взаимодействии с кислородом воздуха.

Любой процесс горения сводится к взаимодействию



Количество горючих веществ составляет $G_{ГВ} = 130$ кг, а тепловой эффект реакции составляет $Q_{\text{тепл.углерода}} = 34,07$ МДж/кг.

Пожарная нагрузка определяется по формуле

$$Q = Q_{\text{тепл.углерода}} \cdot G_{ГВ}, \quad (4.8)$$

$$Q = 130 * 34,07 = 4511 \text{ МДж}$$

Следовательно, пожарная нагрузка Q составляет 4511 МДж. Удельная пожарная нагрузка в соответствии санитарно-эпидемиологическими требованиями определяется по формуле

$$q = Q / S \quad (4.9)$$

где $S = 12\text{м}^2$ – площадь помещения;

$$q = 4511 / 12 = 375$$

Удельная пожарная нагрузка q составляет 375 МДж/м².
Проверяется условие

$$Q < 0,64\text{Ч}_q \text{ Ч}_н \quad (4.10)$$

Это условие не выполняется, следовательно, помещение соответствует категории В2. В качестве средств пожаротушения применяется ручной огнетушитель ОУ-2, содержащий углекислотные компоненты. Исправность огнетушителя должна периодически проверяться.

4.10 Вывод

В этом разделе дипломной работы был произведен анализ условий труда в данном офисном помещении, в частности, расчет электромагнитного воздействия и анализ средств пожаротушения. Приведённые выше расчёты показывают, что использование мониторов соответствует нормам, при которых воздействие электромагнитного поля не несёт ущерб здоровью работника.

Пожарная безопасность не соответствовала норме, что было исправлено путем приобретения огнетушителя.

Заключение

В данном дипломном проекте был разработан программный продукт для исследования свойств пространственной фокусировки осесимметричных электронных линз. Целью разработки данного продукта являлось упрощение решения сложных вычислительных процессов, связанных с электронной оптикой.

Разработанный программный продукт обладает понятным интерфейсом, защитой от человеческого фактора, то есть от ошибок, возникающих в ходе задания начальных параметров системы.

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

Для данного программного продукта были рассчитаны экономическая эффективность, противопожарная безопасность, а также расчет электромагнитного излучения от компьютера, влияющего на человека при работе с ним.

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 ГОСТ 12.1.028-80. Определение шумовых характеристик источников шума.
- 6 СНиП РК 2.04-05-2002. Естественное и искусственное освещение. Общие требования.
- 7 Скала Н.В. Основы организации охраны и безопасности труда в Республике Казахстан. – Алматы, 2007.
- 8 Безопасность жизнедеятельности. Белов С.В., Ильницкая А.В., Козьяков А.Ф. 2007
- 9 Бимурзаев С.Б. Методы расчёта электронных линз: Учебное пособие. – Алматы, 2001. – 52 с.
- 10 Кельман В.М. и Явор С.Я. «Электронная оптика». Л.: Наука, 1968.
- 11 Баранова Л.А. и Явор С.Я. «Электростатические электронные линзы». М.: Наука, 1986.

Приложение А

Код программы:

```
unit Unit1;

interface

uses
    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;
        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;
    end;
end;
```

Продолжение Приложения А

```
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;
```

Продолжение Приложения А

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

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);
```

Продолжение Приложения А

```
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;
```

Продолжение Приложения А

```
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,
```

Продолжение Приложения А

```
    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]));
```

Продолжение Приложения А

```
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;
```


Продолжение Приложения А

```
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;
```

Продолжение Приложения А

```
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}

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

procedure TForm2.Button2Click(Sender: TObject);
```

Продолжение Приложения А

```
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.

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;
```

Продолжение Приложения А

```
{$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)
    Memo1: TMemo;
    Button1: TButton;
    Button2: TButton;
    procedure Button1Click(Sender: TObject);
    procedure Button2Click(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  Form4: TForm4;

implementation

uses Unit2;
```

Продолжение Приложения А

```
{R *.dfm}

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;

{R *.dfm}
```

Окончание Приложения А

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