

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РЕСПУБЛИКИ КАЗАХСТАН

Некоммерческое акционерное общество «Казахский национальный исследовательский
технический университет имени К.И. Сатбаева»



SATBAYEV
UNIVERSITY

Институт автоматки и информационных технологий

Кафедра «Робототехники и технических средств автоматки»

Абрешов Кажымукан Абдуллаулы

«Разработка компьютерного приложения для расчёта резерфордовских сечений»

МАГИСТЕРСКАЯ ДИССЕРТАЦИЯ

7М07107 – Робототехника и мехатроника

Алматы 2023

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РЕСПУБЛИКИ КАЗАХСТАН
Некоммерческое акционерное общество «Казахский национальный исследовательский
технический университет имени К.И. Сатбаева»



SATBAYEV
UNIVERSITY

Институт автоматки и информационных технологий

Кафедра «Робототехники и технических средств автоматки»

Абрешов Кажымуқан Абдуллаұлы


Для получения ученой степени магистра наук

МАГИСТЕРСКАЯ ДИССЕРТАЦИЯ


Название диссертации: «Разработка компьютерного приложения для расчёта
резерфордовских сечений»


Направление подготовки: 7М07107 – Робототехника и мехатроника

Кандидат физико-математических наук
Заведующий кафедрой
Институт автоматки и информационных
технологий

 Алдияров Н.У.
«31» май 2023 ж.

Рецензент
к.т.н., Доктор Ph.D.
Ассоциированный профессор
Институт автоматки и информационных
технологий

 Балбаев Г.К.
«31» май 2023 ж.

Норма контроль
доктор Ph.D.,
ассоциированный профессор
 Бектилев А.Ю.
«31» май 2023 ж.



Алматы 2023

«К.И. Сәтбаев атындағы Қазақ ұлттық техникалық зерттеу университеті» коммерциялық емес акционерлік қоғамы



SATBAYEV
UNIVERSITY

Институт автоматизации и информационных технологий
Кафедра «Робототехники и технических средств автоматизации»

7M07107 – Робототехника и мехатроника



УТВЕРЖДАЮ
Заведующий кафедрой РТиТСА
Кандидат технических наук,
Профессор

К.А. Ожикенов
«31» май 2023 ж.

ЗАДАНИЕ
на выполнение магистерской диссертации

Магистрант Абдрешов Кажымукан Абдуллаулы

Тема: Разработка компьютерного приложения для расчета резерфордовских сечений

Утверждена приказом ректором Университета № 153-24 от «29» 10 2021г.

Срок сдачи законченной диссертации: «31» июнь 2023 ж.

Исходные данные к магистерской диссертации:

Перечень подлежащих разработке в магистерской диссертации вопросов:

- сделать аналитический обзор*
- программирование*
- разработка программного обеспечения*

Перечень графического материала (с точным указанием обязательных чертежей):

Презентация работы показана на 14 слайдах

Рекомендуемая основная литература: 51 библиография и приложение

ГРАФИК
подготовки магистерской диссертации

Наименование перечень вопросов	разделов, разрабатываемых	Сроки представления научному руководителю	Примечание
Технологическая часть		16.01.23	выполнено
Исследовательская часть		21.02.23	выполнено
Часть программирования		13.04.23	выполнено
Заключительная часть		15.05.23	выполнено

Подписи
консультантов и нормоконтролера на законченную магистерскую диссертацию с
указанием относящихся к ним разделов диссертации

Наименования разделов	Консультанты, И.О.Ф. (уч. степень, звание)	Дата подписания	Подпись
Нормоконтролер	доктор PhD, ассоциированный профессор Бектилеов А.Ю	02.06.23	
Главная часть	Кандидат физико- математических наук Алдияров Н.У.	02.06.23	
Раздел расчета	Кандидат физико- математических наук Алдияров Н.У.	02.06.23	

Научный руководитель

Задание принял к исполнению обучающийся

Дата

Алдияров Н.У.

Абрешов К.А.

«31» май 2023 ж.

АНДАТПА

Резерфорд қималарын есептеуге арналған компьютерлік қосымшаны әзірлеу Резерфордтың шашырауы атом мен ядроның құрылымын анықтауға көмектескен негізгі физикалық құбылыстардың бірі болып табылады.

Резерфорд қималарын есептеу ядролық физика және элементар бөлшектер физикасы саласындағы маңызды құрал болып табылады. Бұл рефератта біз Резерфорд қималарын есептеуге арналған компьютерлік қосымшаны жасау процесін қарастырамыз. Мақсаттардың анықтамасы: Әзірлеуді бастамас бұрын қолданбаның мақсаттарын анықтау керек. Негізгі мақсаттардың бірі бөлшектердің атомдық ядролардағы Резерфордтық шашырауын модельдеу және болжау мүмкіндігі. Қолданба пайдаланушыға бөлшектер мен ядролардың параметрлерін енгізуге, сондай-ақ сәйкес Резерфорд қималарын есептеуге мүмкіндік беруі керек. Дереккөздер мен алгоритмдерді зерттеу: Келесі қадам Резерфорд шашырауының физикалық негіздерін және көлденең қималарды есептеудің қол жетімді алгоритмдерін зерттеу болып табылады. Резерфорд қималарын есептеуде қолданылатын математикалық модельдер мен формулаларды түсіну үшін ғылыми мақалаларды, кітаптарды және басқа да дереккөздерді талдау қажет. Бағдарламалау тілі мен құралдарын таңдау: Зерттеуден кейін бағдарламалау тілін және сәйкес әзірлеу құралдарын таңдау қажет. Мысалы, Python ғылыми есептеулерге арналған танымал бағдарламалау тілі болып табылады және NumPy және SciPy сияқты көптеген кітапханаларды ұсынады, олар Резерфордтың шашыраңқы математикалық модельдерін жүзеге асыру үшін пайдалы болуы мүмкін. UI әзірлеу: Компьютерлік қосымшаның пайдаланушыға ыңғайлы және интуитивті интерфейсі болуы керек. Бұл пайдаланушыларға бөлшектер мен ядролардың параметрлерін оңай енгізуге, қажетті опцияларды таңдауға және есептеу нәтижелерін алуға мүмкіндік береді.

АННОТАЦИЯ

Разработка компьютерного приложения для расчета резерфордовских сечений
Резерфордовское рассеяние является одним из фундаментальных физических явлений, которое помогло установить структуру атома и ядра. Расчет резерфордовских сечений является важным инструментом в области ядерной физики и физики элементарных частиц. В данном реферате рассмотрим процесс разработки компьютерного приложения для расчета резерфордовских сечений.

Определение целей:

Перед началом разработки необходимо определить цели приложения. Одной из главных целей является возможность моделирования и предсказания резерфордовского рассеяния частиц на ядрах атомов. Приложение должно предоставлять пользователю возможность ввода параметров частиц и ядер, а также расчета соответствующих резерфордовских сечений.

Исследование источников и алгоритмов:

Следующим шагом является изучение физических основ резерфордовского рассеяния и доступных алгоритмов для расчета сечений. Необходимо провести анализ научных статей, книг и других источников, чтобы понять математические модели и формулы, используемые в расчетах резерфордовских сечений.

Выбор языка программирования и инструментов:

После исследования необходимо выбрать язык программирования и соответствующие инструменты разработки. Например, Python является популярным языком программирования для научных вычислений и предоставляет множество библиотек, таких как NumPy и SciPy, которые могут быть полезны для реализации математических моделей резерфордовского рассеяния.

Разработка пользовательского интерфейса:

Компьютерное приложение должно иметь удобный и интуитивно понятный пользовательский интерфейс. Это позволит пользователям легко вводить параметры частиц и ядер, выбирать необходимые опции и получать результаты расчетов.

ABSTRACT

Development of a computer application for calculating Rutherford cross sections
Rutherford scattering is one of the fundamental physical phenomena that helped to establish the structure of the atom and the nucleus. The calculation of Rutherford cross sections is an important tool in the field of nuclear physics and elementary particle physics. In this abstract, we will consider the process of developing a computer application for calculating Rutherford cross sections.

Definition of goals:

Before you start developing, you need to define the goals of the application. One of the main goals is the possibility of modeling and predicting the Rutherford scattering of particles on atomic nuclei. The application should allow the user to enter the parameters of particles and nuclei, as well as calculate the corresponding Rutherford cross sections.

Research of sources and algorithms:

The next step is to study the physical foundations of Rutherford scattering and available algorithms for calculating cross sections. It is necessary to analyze scientific articles, books and other sources in order to understand the mathematical models and formulas used in the calculation of Rutherford cross sections.

Choice of programming language and tools:

After the research, it is necessary to choose a programming language and appropriate development tools. For example, Python is a popular programming language for scientific computing and provides many libraries such as NumPy and SciPy that can be useful for implementing Rutherford scattering mathematical models.

UI development:

The computer application should have a user-friendly and intuitive user interface. This will allow users to easily enter the parameters of particles and nuclei, select the necessary options and receive calculation results.

СОДЕРЖАНИЕ

ВВЕДЕНИЕ

1 Технологическая часть

1.1 Ядерная физика

1.2 Ядерная реакция

1.3 Реактивные частицы

1.4 Ускоритель. Изохронный циклотрон У-150М

2 Исследовательская часть

2.1 Законы сохранения энергии и импульса

2.2 Кинематика процессов рассеяния частиц в ядрах

2.3 Описание процесса рассеяния в лабораторной системе

2.4 Упругое рассеяние частиц

2.5 Диаграмма импульса рассеяния

3 Часть программирования

3.1 Программирование на языке Python

3.2 Графический интерфейс программы

3.3 Анализ различных спектров

Заключение

Список литературы

Приложение А

ВВЕДЕНИЕ

Резерфордовские сечения (также известные как дифференциальные или полные сечения рассеяния Резерфорда) являются важной характеристикой взаимодействия частиц с ядрами атомов. Они представляют собой вероятность рассеяния частицы на ядре под определенным углом.

В экспериментах Резерфорда проводился рассеяние альфа-частиц на тонких фольгах, таких как золото или платина. Он наблюдал отклонение альфа-частиц под разными углами, что привело к открытию ядра атома и формулировке модели атома, в которой положительный заряд сконцентрирован в малом и плотном ядре.

Резерфордовские сечения можно рассчитать с использованием теории рассеяния, основанной на принципах классической или квантовой механики. В классической модели Резерфорда рассеяние рассматривается как кулоновское взаимодействие между заряженными частицами - альфа-частицей и ядром. В квантовой механике рассматривается волновая функция частицы и использование уравнений Шредингера для расчета вероятности рассеяния под определенным углом.

Расчет резерфордовских сечений включает учет различных параметров, таких как энергия частицы, заряд ядра, масса частицы и ядра, угол рассеяния и другие. Для сложных систем существуют различные методы расчета, включая аналитические и численные методы, которые позволяют получить точные результаты.

Резерфордовские сечения имеют широкое применение в физике элементарных частиц, ядерной физике, астрофизике и других областях. Они играют важную роль в понимании структуры атомных ядер, разработке ядерных реакторов, лучевой терапии и других технологиях, связанных с исследованием и использованием атомного ядра.

1 . ТЕХНОЛОГИЧЕСКАЯ ЧАСТЬ

1.1 Ядерная физика

Резерфордовские сечения являются важным понятием в ядерной физике и используются для описания вероятности рассеяния частиц на ядрах атомов. Они названы в честь известного физика Эрнеста Резерфорда, который внес значительный вклад в изучение атомной структуры.

Резерфордовское сечение обозначает вероятность рассеяния частицы на ядре под определенным углом. Оно измеряется в единицах площади, и его значения зависят от энергии и типа рассеивающей частицы, а также от ядра, на котором происходит рассеяние.

Самое известное резерфордовское сечение - это сечение Кулона, которое описывает рассеяние заряженных частиц на точечном заряде, таком как протон или ядро. Сечение Кулона выражается через параметры зарядов и энергию рассеяния, и оно показывает, как вероятность рассеяния частицы зависит от угла рассеяния и энергии.

Другие формы резерфордовских сечений включают дифференциальное сечение, интегральное сечение и тотальное сечение. Дифференциальное сечение определяет вероятность рассеяния частицы в определенном направлении и угле, интегральное сечение представляет собой сумму вероятностей рассеяния для всех возможных углов, а тотальное сечение представляет собой суммарную вероятность рассеяния во всех направлениях.

Резерфордовские сечения имеют широкий спектр применений, особенно в ядерной физике, астрофизике и радиационных науках. Они используются для изучения взаимодействия частиц с ядрами, определения структуры ядра, анализа радиоактивных материалов и моделирования ядерных реакций. Также они играют важную роль в разработке методов диагностики и лечения рака, в проектировании ядерных реакторов и в исследованиях фундаментальных свойств материи.

Изучение резерфордовских сечений позволяет получить информацию о структуре и свойствах атомных ядер, таких как их размеры, формы и зарядовое распределение. Резерфордовское рассеяние используется для измерения радиусов ядер и определения зарядовых плотностей в ядрах.

В экспериментах по резерфордовскому рассеянию обычно используются заряженные частицы, такие как протоны или альфа-частицы, которые падают на образец, содержащий ядра. При рассеянии частицы испытывают взаимодействие с электрическим полем ядра, что приводит к отклонению и изменению их траектории. Измеряя углы рассеяния и распределение интенсивности рассеянных частиц, можно получить информацию о структуре ядра.

Резерфордовское рассеяние также используется для исследования ядерных реакций и свойств элементарных частиц. Например, в экспериментах на ускорителях частиц используют резерфордовское рассеяние для изучения столкновений высокоэнергетических частиц и определения характеристик взаимодействия.

Резерфордовские сечения также находят применение в медицине и радиационной терапии. Используя знание резерфордовских сечений, можно расчетно определить дозу облучения при проведении радиотерапии для лечения рака и минимизировать воздействие на окружающие здоровые ткани.

Основными методами анализа резерфордовского рассеяния являются определение энергии и угла рассеяния частиц, а также измерение интенсивности рассеяния в зависимости от угла и энергии. Для более точного и детального изучения ядерной структуры применяются математические модели и теории, основанные на законах электромагнетизма и квантовой механике.

Ядерная физика является областью физики, которая изучает структуру, свойства и взаимодействия ядер - центральных частей атомов. Она изучает ядра как системы, состоящие из протонов и нейтронов, называемых нуклонами, и их взаимодействие с другими частицами и излучением.

Основные темы исследования в ядерной физике включают:

Структура ядра: Ядра состоят из протонов, нейтронов и других частиц, которые образуют ядерные оболочки. Изучение структуры ядра включает определение их массы, размеров, формы и энергетических уровней. Используя различные экспериментальные методы, такие как резерфордовское рассеяние, спектроскопия и ядерная резонансная релаксация, и теоретические модели, ядерные физики исследуют внутреннюю структуру и свойства ядер.

Ядерные реакции: Ядерные реакции происходят, когда ядра сталкиваются и взаимодействуют друг с другом или с другими частицами. Эти реакции могут приводить к изменению состава ядра, испусканию частиц или излучению. Исследование ядерных реакций включает изучение кинематики и энергетики взаимодействующих частиц, а также измерение и моделирование вероятностей и сечений реакций.

Ядерные силы и модели: Ядерные силы играют важную роль во взаимодействии нуклонов в ядрах. Ядерные физики разрабатывают различные теории и модели, такие как модель жидкой капли, модель оболочки и квантовохромодинамические модели, для объяснения и предсказания свойств ядерных систем.

Ядерная энергетика: Ядерная физика также связана с применением ядерных реакций в производстве энергии. Ядерные реакторы используются для генерации электроэнергии, основанной на делении ядер тяжелых элементов или слиянии.

Ядерная медицина: Ядерная физика имеет важное применение в медицине. Радиоактивные изотопы используются в диагностических процедурах, таких как позитронно-эмиссионная томография (ПЭТ) и радионуклидная диагностика, а также в радиотерапии для лечения рака. Ядерные методы обладают высокой чувствительностью и позволяют получить детальные изображения внутренних органов и тканей, а также достичь точной и таргетированной терапии.

Космическая ядерная физика: Изучение ядерной физики играет важную роль в исследовании космического пространства и астрофизике. Ядерные реакции и процессы, происходящие в звездах, суперновых и других астрономических объектах, а также в космическом излучении, изучаются для понимания

формирования и эволюции Вселенной.

Ядерная безопасность и нераспространение: Ядерная физика также связана с вопросами ядерной безопасности, контроля ядерных материалов и предотвращения распространения ядерного оружия. Методы анализа ядерных материалов, такие как спектрометрия гамма-излучения и изотопный анализ, применяются для обнаружения и контроля ядерных материалов и наблюдения за возможными нарушениями ядерных договоров. Ядерная физика является важной и интересной областью науки, которая охватывает широкий спектр тем и применений. Она имеет значение для нашего понимания фундаментальных законов природы, а также для развития новых технологий и применений в различных областях, таких как энергетика, медицина и космические исследования.

1.2 Ядерная реакция

Ядерная реакция - это процесс, в котором происходят изменения в ядрах атомов. Ядерные реакции могут быть спонтанными, такими как радиоактивный распад, или инициироваться в результате столкновения ядер с другими частицами.

В ядерных реакциях происходит изменение состава ядра, что приводит к образованию новых ядер и/или испусканию частиц и энергии. Ядерные реакции обычно сопровождаются переходом ядер на более стабильные состояния, а также высвобождением энергии в форме тепла, света или радиоактивного излучения. Ядерные реакции классифицируются по типу взаимодействия частиц, вовлеченных в реакцию. Некоторые из основных типов ядерных реакций включают:

Расщепление ядра (фиссия): В этом типе реакции ядро разделяется на две или более более легких частицы при столкновении с другой частицей или абсорбировании нейтрона. Это явление играет ключевую роль в ядерных реакторах и атомных бомбах.

Слияние ядер (фьюзия): В этом типе реакции два или более легких ядра объединяются, образуя более тяжелое ядро. Фьюзия происходит при высоких температурах и давлениях, как внутри звезд, и является источником энергии в термоядерных реакциях.

Изотопические реакции: Это реакции, в которых ядра одного элемента претерпевают изменения в количестве нейтронов или протонов. Примером может служить переход ядра урана-235 в плутоний-239 при поглощении нейтрона.

Выбивание частиц (захват или эмиссия): В этом типе реакции ядро захватывает или эмитирует частицы, такие как протоны, нейтроны, альфа-частицы или другие ядра. Эти реакции играют важную роль в радиоактивном распаде и в процессе синтеза новых элементов.

Ядерные реакции имеют несколько характеристик, которые важны для их описания и понимания:

Сечение поглощения: Это вероятность того, что ядро поглотит частицу или поглощающий поток частиц. Сечение поглощения зависит от энергии

поглощающей частицы и свойств ядра. Оно играет важную роль в анализе радиоактивных материалов и проектировании ядерных реакторов.

Период полураспада: Это время, в течение которого половина ядерных частиц вещества распадается. Различные ядра имеют разные периоды полураспада, что определяет их стабильность или радиоактивность.

Энергетический выход: Это количество энергии, которое высвобождается или поглощается в результате ядерной реакции. Энергетический выход может быть в форме тепла, света, радиоактивного излучения и других форм энергии.

Закон сохранения энергии и импульса: Ядерные реакции должны соблюдать законы сохранения энергии и импульса. Это означает, что суммарная энергия и импульс до и после реакции должны быть одинаковыми.



Рисунок 1.2 – Планетарная модель Резерфорда

Ядерные реакции имеют широкий спектр применений и являются основой для множества технологий и исследований. Они используются в ядерной энергетике для производства электроэнергии, в медицине для диагностики и лечения рака, в науке для исследования структуры и свойств атомных ядер, а также в создании новых элементов в ядерной синтезе.

Влияние энергии и угла падения: Энергия и угол падения влияют на характер реакции и процесс рассеяния. При более высоких энергиях и больших углах падения может происходить глубокое проникновение частиц в ядро, что приводит к более интенсивному взаимодействию и возможным ядерным реакциям.

Процессы перезарядки и передачи энергии: В ядерных реакциях может происходить передача энергии и перезарядка между частицами. Например, в реакциях слияния ядер в звездах происходит передача энергии от одного ядра к другому, что позволяет преодолеть электростатический барьер и объединиться.

Эффекты возбуждения и испускания излучения: Ядерные реакции могут приводить к возбуждению ядерных состояний, которые затем могут испускать излучение в виде гамма-квантов или других форм излучения. Эти процессы

позволяют изучать структуру ядра и определять его энергетические уровни.

Множественные реакции и ветвление: В некоторых случаях ядерные реакции могут происходить параллельно в различных каналах, приводя к образованию различных продуктов. Это явление известно как множественные реакции или ветвление и требует учета при анализе и моделировании ядерных процессов.

Ядерные реакции играют важную роль во многих аспектах нашей жизни, от производства энергии до медицинской диагностики и исследований фундаментальной физики. Понимание этих реакций и их анализ позволяют нам использовать ядерную энергию безопасно и эффективно, разрабатывать новые методы диагностики и лечения заболеваний, а также расширять наше знание о строении и свойствах атомного мира.

1.3 Реактивные частицы

Реактивные частицы – это частицы, которые обладают свойством активно взаимодействовать с другими веществами или частицами в реакциях химической или физической природы. Они обычно характеризуются высокой энергией или нестабильностью, что позволяет им быстро реагировать и изменять свою структуру или состояние.

Реактивные частицы могут быть различного типа в зависимости от контекста и области науки. Например, в химии реактивные частицы могут быть ионами, радикалами или молекулами с открытыми химическими связями. Они обладают некомплектным электронным окружением и могут активно взаимодействовать с другими веществами, образуя новые химические соединения.

В физике реактивные частицы могут быть высокоэнергетическими частицами, такими как электроны, протоны, альфа-частицы или другие элементарные частицы. Они обычно используются в ускорителях частиц для изучения структуры вещества и фундаментальных взаимодействий. Такие частицы могут сталкиваться с другими частицами, вызывая различные процессы, такие как рассеяние, рождение новых частиц или ядерных реакций.

Реактивные частицы также могут быть связаны с окружающей средой и реакциями, происходящими в атмосфере или в биологических системах. Например, в атмосферной химии реактивные частицы, такие как кислородные радикалы или азотные оксиды, играют важную роль в реакциях, приводящих к образованию озона или других загрязнений. В биологии реактивные частицы, такие как свободные радикалы, могут быть связаны с окислительным стрессом и различными патологическими процессами в организме.

В лабораторных условиях исследования реактивных частиц проводятся с использованием различных методов и техник. Спектроскопические методы, такие как инфракрасная спектроскопия, УФ-видимая спектроскопия и ядерный магнитный резонанс, позволяют анализировать структуру и свойства реактивных частиц. Методы масс-спектрометрии позволяют

определить массу и состав частиц, а методы рентгеновской дифракции – изучать их кристаллическую структуру.

Также существуют различные модели и компьютерные программы, которые помогают предсказывать поведение реактивных частиц, их реакционную способность и влияние на окружающую среду. Это позволяет улучшить процессы синтеза веществ, разработать новые катализаторы с повышенной эффективностью и безопасностью, а также оценить возможные последствия выбросов реактивных частиц в окружающую среду.

Изучение реактивных частиц имеет применение в различных областях, таких как химическая промышленность, фармацевтика, энергетика, экология, материаловедение и многие другие. Понимание и контроль реактивных частиц позволяют разрабатывать более эффективные и экологически безопасные процессы и технологии, а также предотвращать и устранять негативные последствия их воздействия на окружающую среду и здоровье человека.

В целом, изучение реактивных частиц является важным направлением научных исследований, которое способствует развитию различных отраслей науки и технологий, а также помогает решать актуальные проблемы современного мира, связанные с энергетикой, окружающей средой и здоровьем.

1. 4 Ускоритель. Изохронный циклотрон У-150М

Изохронный циклотрон У-150М - это тип ускорителя заряженных частиц, который используется для ускорения ионов до высоких энергий. Он является одним из вариантов циклотрона, который был разработан в Институте ядерной физики имени Г. И. Будкера Сибирского отделения Российской академии наук.

Принцип работы изохронного циклотрона основан на использовании переменного магнитного поля, которое позволяет поддерживать постоянную фазу ускоряемых частиц на протяжении всего ускорительного цикла. Это позволяет достичь высокой эффективности ускорения частиц и достигнуть значительных энергий.

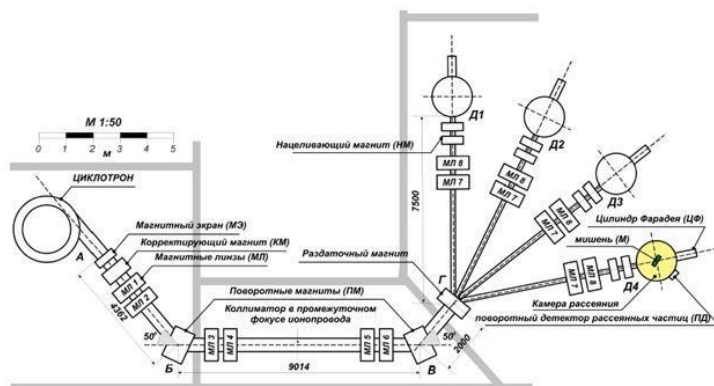
Ускоритель У-150М имеет диаметр магнитной системы в 2 метра и может ускорять протоны до энергий порядка 150 МэВ. Он состоит из главного магнитного поля, радиально-симметричной электростатической системы и многосекционного ускорительного пространства.

Процесс ускорения в изохронном циклотроне происходит следующим образом: заряженные частицы, например протоны, вводятся в ускоритель и начинают движение в магнитном поле, которое создается главным магнитом. Под действием этого магнитного поля частицы начинают двигаться по спиральной траектории, при этом каждый оборот они проходят через ускорительное пространство, где электрическое поле дает им энергию ускорения. Постепенно частицы увеличивают свою энергию и радиус траектории, пока не достигнут заданной энергии.

Важной особенностью изохронного циклотрона У-150М является возможность

ускорения не только протонов, но и других заряженных частиц, таких как дейтроны, альфа-частицы и другие ионы. Для этого необходимо соответствующим образом настроить магнитное поле и электрическое поле в ускорительном пространстве.

Ускоритель У-150М применяется в различных областях исследований, таких



как

Рисунок 1.4 – Рассеяние пучка прогрессивных ионов из циклотронной камеры передача схемы на камеру

ядерная физика, физика элементарных частиц, медицинская физика и другие. Он используется для проведения экспериментов, связанных с изучением взаимодействия заряженных частиц с веществом, получения пучков высокоэнергетических частиц для исследования структуры ядра, создания изотопов для медицинских целей и многого другого.

Изохронные циклотроны, включая У-150М, имеют ряд преимуществ, таких как относительная простота конструкции, компактность и низкая стоимость по сравнению с другими типами ускорителей, такими как синхротроны или линейные ускорители. Они также обладают достаточно высокой интенсивностью пучков, что позволяет проводить эксперименты с высокой статистической точностью.

Однако изохронные циклотроны имеют и некоторые ограничения. Они ограничены максимальной достижимой энергией частиц, что делает их менее подходящими для исследования физики высоких энергий. Также важным аспектом является эффективность ускорения частиц, которая ограничена физическими параметрами ускорителя и возможностью поддерживать стабильность и точность ускорительного процесса.

В целом, изохронный циклотрон У-150М представляет собой важное ускорительное устройство, позволяющее ускорять заряженные частицы до высоких энергий.

Изохронный циклотрон У-150М имеет ряд преимуществ, которые делают его востребованным в научных и прикладных областях. Вот некоторые из них:

1. Высокая энергетическая эффективность: У-150М способен ускорять частицы до значительных энергий, достигая порядка 150 МэВ. Это позволяет проводить исследования на высоких энергетических уровнях и получать значимые результаты.

2. Гибкость в выборе частиц: У-150М может работать с различными типами

заряженных частиц, включая протоны, дейтроны и другие ионы. Это делает его универсальным инструментом для исследований в различных областях науки.

3. Компактность и относительная простота конструкции: У-150М является относительно компактным ускорителем, что делает его более доступным и удобным для установки в лабораториях. Он также имеет относительно простую конструкцию, что упрощает его эксплуатацию и обслуживание.

4. Экспериментальные возможности: Использование У-150М позволяет проводить разнообразные эксперименты в ядерной физике, физике элементарных частиц и других областях. Это включает изучение структуры ядра, взаимодействия частиц с веществом, создание изотопов для медицинских и промышленных целей и многое другое.

5. Прикладные применения: У-150М находит применение в медицине, атомной энергетике, материаловедении и других областях. Он может использоваться для лечения опухолей при использовании метода радиотерапии, создания изотопов для диагностики и лечения, исследования новых материалов с использованием ионной имплантации и других технологий.

Инновационность: Ускоритель У-150М является продуктом современных научных и инженерных разработок. Он постоянно совершенствуется и модернизируется для достижения более высоких энергий, улучшения стабильности и повышения эффективности ускорения.

Ускоритель У-150М, как и другие изохронные циклотроны, играет важную роль в различных областях науки и технологий. Вот некоторые дополнительные аспекты, которые следует упомянуть:

Исследование фундаментальных вопросов: Ядерная физика и физика элементарных частиц являются ключевыми областями исследования, где ускорители, включая У-150М, играют важную роль. Они помогают раскрыть тайны структуры атомного ядра, взаимодействий элементарных частиц и фундаментальных сил природы.

Развитие новых материалов и технологий: Ускорители используются для создания новых материалов с помощью ионной имплантации, которая позволяет изменять свойства поверхности материалов или создавать новые функциональные покрытия. Это имеет широкие применения в электронике, оптике, нанотехнологиях и других областях.

Медицинская диагностика и лечение: Использование ускорителей, таких как У-150М, в медицине позволяет проводить диагностику и лечение различных заболеваний. Они используются для создания изотопов, используемых в ядерной медицине для диагностики опухолей, исследования функционирования органов и терапии рака.

Промышленные исследования и производство: Ускорители также находят применение в промышленных исследованиях и производстве. Они могут использоваться для модификации материалов, создания новых покрытий и обработки поверхностей. Также ускорители могут быть вовлечены в различные технологические процессы, такие как воздействие на материалы высокоэнергетическими частицами или ионной имплантацией для улучшения их

свойств.

Благодаря продвинутым ускорителям, таким как У-150М, исследователи могут изучать физические процессы на самых высоких энергетических уровнях, открывая новые пути для понимания основ физики и расширения наших знаний о Вселенной. Это также помогает разрабатывать новые методы диагностики и лечения заболеваний, создавать более эффективные и инновационные материалы и технологии, а также содействует прогрессу в области энергетики, промышленности и других отраслях.

Ускорители, такие как У-150М, являются ключевыми инструментами для исследования микромира и расширения границ нашего понимания физических процессов. Они играют важную роль в научных открытиях, инженерных разработках и инновациях, открывая новые возможности для развития науки, технологий и общества в целом.

Таким образом, У-150М и подобные ускорители являются фундаментальными инструментами в научных исследованиях, медицине, промышленности и других областях, вносящими существенный вклад в развитие человечества и современного мира.

2 ИССЛЕДОВАТЕЛЬСКАЯ ЧАСТЬ

2.1 Законы сохранения энергии и импульса

Законы сохранения энергии и импульса являются основополагающими принципами в физике и играют важную роль в описании различных физических процессов. Они помогают определить, как энергия и импульс распределяются и сохраняются в системах, подверженных взаимодействиям.

Закон сохранения энергии гласит, что в изолированной системе энергия остается неизменной. Это означает, что энергия не может быть создана или уничтожена, а только преобразована из одной формы в другую. В системе, где отсутствуют внешние силы или потери энергии, сумма кинетической энергии и потенциальной энергии остается постоянной.

Закон сохранения импульса утверждает, что в изолированной системе импульс остается постоянным. Импульс определяется массой и скоростью тела и является векторной величиной. Взаимодействия между телами в системе приводят к обмену импульсом, но сумма импульсов всех тел в системе остается неизменной.

Эти два закона взаимосвязаны и тесно связаны друг с другом. Закон сохранения энергии можно интерпретировать как следствие закона сохранения импульса, и наоборот. Например, при взаимодействии тел в системе сила, действующая на одно тело, вызывает изменение его импульса, а изменение кинетической энергии этого тела связано с работой, выполненной этой силой. Эти законы применяются в различных областях физики, таких как механика, электродинамика, термодинамика и т.д. Они позволяют проводить анализ и прогнозирование физических явлений и процессов, а также разрабатывать и оптимизировать различные технологии и системы.

Законы сохранения энергии и импульса являются фундаментальными принципами, которые позволяют нам понять и объяснить поведение физических систем. Они являются универсальными и применимыми на всех уровнях рассмотрения - от элементарных частиц до макроскопических систем. Эти законы являются основой для построения математических моделей и уравнений, описывающих движение и взаимодействия объектов в природе.

Закон сохранения энергии позволяет нам анализировать и предсказывать изменение энергетического состояния системы. Он находит применение во многих областях физики, начиная от механики, где мы можем определить изменение кинетической и потенциальной энергии тела, до термодинамики, где закон сохранения энергии выражается в законе сохранения энергии в тепловых процессах. Закон сохранения импульса имеет большое значение при рассмотрении движения тел и взаимодействий между ними. Он позволяет определить изменение импульса системы в результате взаимодействия, а также прогнозировать и анализировать движение тел в различных условиях. Закон сохранения импульса применяется в механике, электродинамике, оптике и других областях физики.

Вместе эти два закона обеспечивают консистентность и взаимосвязь между различными аспектами физических систем. Они позволяют нам проводить анализ и моделирование сложных явлений, таких как столкновения частиц, движение тел в поле силы или преобразование энергии в различных процессах.

Применение законов сохранения энергии и импульса распространено не только в науке, но и в различных технических областях. Они используются для проектирования и оптимизации механических систем, электронных устройств, энергетических систем и других технологий. Соблюдение этих законов позволяет обеспечить эффективность и стабильность работы систем.



Рисунок 2.1 – Закон сохранения импульса

Закон сохранения энергии имеет фундаментальное значение во всех областях физики. Он утверждает, что энергия не может быть создана или уничтожена, а только преобразована из одной формы в другую. Это означает, что сумма кинетической энергии, потенциальной энергии и других форм энергии в изолированной системе остается постоянной со временем. Например, при движении тела под действием силы его кинетическая энергия может увеличиваться, в то время как потенциальная энергия может уменьшаться, но сумма этих двух энергий остается постоянной.

Закон сохранения импульса утверждает, что в изолированной системе сумма импульсов всех взаимодействующих тел остается постоянной. Импульс определяется массой и скоростью тела и является векторной величиной. При взаимодействии двух тел силы, действующие между ними, вызывают равные и противоположно направленные изменения импульса каждого тела, так что их сумма остается неизменной. Например, при столкновении двух тел, если одно тело приобретает положительный импульс, то другое тело приобретает равный по величине, но противоположный по направлению импульс.

Законы сохранения энергии и импульса являются взаимосвязанными и взаимодополняющими. Они позволяют проводить анализ и прогнозирование различных физических процессов и явлений, включая движение тел, столкновения, распады частиц, электромагнитные взаимодействия и многое другое. Без этих законов было бы невозможно понять и объяснить множество явлений, с которыми мы сталкиваемся в ежедневной жизни и в научных

исследованиях.

Применение законов сохранения энергии и импульса имеет широкий спектр применений в различных областях физики и техники.

В механике, законы сохранения энергии и импульса играют ключевую роль при анализе движения тел и столкновений. Они позволяют определить конечные скорости и траектории тел после столкновений, а также рассчитать необходимую энергию для преодоления сил сопротивления и поддержания постоянной скорости.

В термодинамике, закон сохранения энергии применяется для анализа тепловых процессов. Он позволяет определить изменение внутренней энергии системы, работу, выполненную над системой или совершенную ею, и теплообмен между системой и окружающей средой.

В электродинамике, законы сохранения энергии и импульса применяются для анализа электромагнитных полей и процессов. Они позволяют определить энергию, переносимую электромагнитными волнами, и импульс, передаваемый заряженными частицами в электромагнитном поле.

В ядерной физике, законы сохранения энергии и импульса играют важную роль при изучении ядерных реакций и распадов. Они позволяют определить энергию и импульс, переносимые ядерными частицами, и рассчитать вероятность различных ядерных процессов.

В технических областях, законы сохранения энергии и импульса используются при проектировании и оптимизации различных систем и устройств. Например, при разработке автомобилей учитывается сохранение энергии при движении и эффективное использование топлива. В энергетике применяются принципы сохранения энергии для создания эффективных и устойчивых источников энергии.

В целом, законы сохранения энергии и импульса являются фундаментальными принципами, которые применяются во всех областях физики и техники. Они позволяют анализировать и предсказывать поведение систем, рассчитывать энергетические и динамические характеристики систем, и создавать эффективные и устойчивые технологии.

Законы сохранения энергии и импульса также имеют фундаментальное значение в контексте фундаментальных теорий физики, таких как общая теория относительности и квантовая механика. В общей теории относительности закон сохранения энергии связан с принципом эквивалентности массы и энергии, выраженным знаменитой формулой $E=mc^2$, где E - энергия, m - масса, c - скорость света. В квантовой механике законы сохранения энергии и импульса являются основой для формулирования принципа неопределенности Гейзенберга, который ограничивает точность одновременного измерения энергии и импульса частицы.

Законы сохранения энергии и импульса являются фундаментальными и всеобъемлющими принципами физики. Они помогают нам понять и объяснить, как энергия и импульс распределяются и сохраняются в различных системах и процессах. Эти законы позволяют нам проводить анализ, прогнозирование и оптимизацию различных физических явлений и технических систем. Без них

было бы невозможно развитие науки и технологий, которые опираются на понимание и контроль физических процессов.

В итоге, законы сохранения энергии и импульса являются универсальными и неотъемлемыми принципами физики. Они играют важную роль в наших усилиях по пониманию и описанию физического мира и служат основой для развития новых технологий и достижений в науке.

2.2 Кинематика процессов рассеяния частиц в ядрах

Кинематика процессов рассеяния частиц в ядрах является важной областью исследований в ядерной физике. Рассеяние частиц - это взаимодействие двух или более частиц, при котором они меняют направление и/или энергию своего движения. В ядерной физике, рассеяние частиц используется для изучения структуры ядер и свойств ядерных сил.

Кинематика процессов рассеяния частиц описывает движение и взаимодействие частиц в процессе рассеяния. Основные величины, характеризующие кинематику рассеяния, включают углы рассеяния, энергии и импульсы частиц до и после рассеяния, а также сечение рассеяния.

Углы рассеяния определяются относительным изменением направления движения частицы после взаимодействия. В рассеянии частиц в ядрах обычно измеряются углы относительно направления падающей частицы. Распределение углов рассеяния может дать информацию о внутренней структуре ядра и взаимодействии между частицами.

Энергия частицы до и после рассеяния также играет важную роль в кинематике рассеяния. Изменение энергии частицы после рассеяния связано с передачей энергии от одной частицы к другой или с поглощением или испусканием фотона. Измерение изменения энергии позволяет определить свойства ядерных сил и взаимодействия между частицами.

Импульс частицы также может изменяться при рассеянии. Изменение импульса связано с взаимодействием частиц и передачей импульса от одной частицы к другой. Измерение изменения импульса позволяет определить характер взаимодействия и структуру ядра.

Сечение рассеяния является вероятностью рассеяния частицы на ядре. Оно определяет вероятность того, что взаимодействие между частицей и ядром приведет к рассеянию, а не к другим типам взаимодействий, таким как поглощение или захват. Сечение рассеяния зависит от энергии падающей частицы, угла рассеяния и свойств ядра. Измерение сечения рассеяния позволяет получить информацию о структуре ядра и взаимодействии между частицами.

Кинематика процессов рассеяния частиц в ядрах может быть изучена с помощью различных методов и экспериментальных установок. Одним из таких методов является рассеяние частиц на фиксированной мишени, при котором падающие частицы сталкиваются с ядрами мишени и рассеиваются под различными углами. Другим методом является рассеяние частиц с использованием акселераторов, где частицы ускоряются до высоких энергий и сталкиваются с мишенью для изучения их рассеяния.

Исследование кинематики процессов рассеяния частиц в ядрах имеет важное значение для понимания структуры и свойств ядер, а также для развития ядерных технологий и применений. Это позволяет получить информацию о внутренних составляющих ядер, таких как нуклонные дистрибуции и взаимодействия между нуклонами, и исследовать различные ядерные процессы, включая ядерные реакции и распады.

Таким образом, кинематика процессов рассеяния частиц в ядрах является важной областью исследований в ядерной физике. Она позволяет изучать структуру ядер и взаимодействия между частицами, а также имеет широкий спектр применений в ядерных и физических науках, от фундаментальных исследований до практических приложений.

Исследования кинематики процессов рассеяния частиц в ядрах позволяют получить ценную информацию о структуре ядер и их взаимодействиях. Кинематические характеристики рассеяния, такие как углы, энергии и импульсы частиц, являются ключевыми параметрами, которые позволяют анализировать процессы рассеяния и извлекать информацию о свойствах ядер и их составляющих.

Одним из основных типов рассеяния является упругое рассеяние, при котором кинетическая энергия и импульс частицы сохраняются до и после взаимодействия. Это означает, что энергия и импульс передаются от одной частицы к другой без потерь. Изучение упругого рассеяния позволяет получить информацию о размере и форме ядра, а также о его нуклонной структуре.

Кроме того, кинематические законы позволяют определить различные типы рассеяния, такие как неупругое рассеяние, при котором происходят изменения внутренней структуры частицы или ядра. Неупругое рассеяние может привести к возбуждению ядра или к его распаду на другие частицы. Изучение неупругого рассеяния дает информацию о возможных состояниях и распадах ядер.

Для изучения кинематики рассеяния частиц в ядрах используются различные экспериментальные методы и установки. Например, метод рассеяния на фиксированной мишени позволяет изучать зависимость углов рассеяния от энергии падающих частиц. Акселераторы используются для ускорения частиц до высоких энергий и проведения более точных экспериментов по рассеянию.

Кинематика процессов рассеяния частиц в ядрах имеет широкий спектр применений. Она находит применение в фундаментальных исследованиях, таких как изучение структуры ядер и ядерных сил, а также в приложениях, связанных с ядерной энергетикой, медициной и другими областями. В ядерной энергетике кинематика рассеяния используется для изучения реакций деления и синтеза ядер, а также для оптимизации процессов ядерного сгорания и производства энергии.

В медицине кинематика рассеяния применяется, например, для разработки методов облучения опухолей при лучевой терапии. Анализ кинематических параметров рассеяния позволяет определить оптимальные условия облучения, минимизирующие повреждение здоровых тканей и максимизирующие эффективность лечения.

Кроме того, кинематика рассеяния частиц в ядрах применяется в астрофизике для изучения ядерных реакций, происходящих в звездах и других космических объектах. Анализ кинематических данных позволяет получить информацию о происходящих ядерных процессах и свойствах звезд.

Исследования кинематики процессов рассеяния частиц в ядрах проводятся с использованием различных методов и инструментов, включая акселераторы частиц, детекторы и анализаторы. Современные экспериментальные техники позволяют получать более точные и детальные данные о кинематике рассеяния и использовать их для более глубокого понимания структуры ядер и их взаимодействий.

Таким образом, кинематика процессов рассеяния частиц в ядрах является важной областью исследований в ядерной физике. Она позволяет изучать свойства ядер и их взаимодействия, а также находит применение в различных областях, включая ядерную энергетику, медицину и астрофизику. Непрерывное развитие экспериментальных методов и технологий позволяет получать все более точные и полные данные о кинематике рассеяния и расширять наше понимание ядерной физики.

В кинематике процессов рассеяния частиц в ядрах важную роль играют различные кинематические величины, которые позволяют описать движение и взаимодействие частиц. Одной из основных величин является угол рассеяния, который определяет изменение направления движения частицы после взаимодействия с ядром. Угол рассеяния может быть измерен экспериментально и использован для извлечения информации о структуре и свойствах ядра. Другой важной кинематической величиной является энергия рассеяния, которая определяет изменение кинетической энергии частицы после рассеяния. Измерение энергии рассеяния позволяет изучать энергетические уровни и возбуждения ядра, а также оценить эффективность передачи энергии от падающей частицы к ядру.

Импульс также играет важную роль в кинематике рассеяния частиц. Изменение импульса частицы после рассеяния может быть использовано для определения массы и импульса ядра. Измерение импульса рассеянных частиц позволяет извлекать информацию о внутренней структуре ядра и о характере взаимодействия между частицами. Важно отметить, что в кинематике процессов рассеяния соблюдаются законы сохранения энергии и импульса. Согласно закону сохранения энергии, сумма кинетической энергии падающей частицы и ядра до взаимодействия равна сумме их кинетической энергии после взаимодействия. Закон сохранения импульса гласит, что сумма импульсов падающей частицы и ядра до взаимодействия равна сумме их импульсов после взаимодействия.

Изучение кинематики процессов рассеяния частиц в ядрах позволяет расширить наше знание о структуре ядра и его взаимодействиях с другими частицами. Это имеет важное значение как в фундаментальных исследованиях, таких как изучение строения и свойств элементарных частиц и ядер, так и в прикладных областях, например, в ядерной энергетике и медицине. Кинематика процессов рассеяния частиц позволяет получить информацию о внутренней

структуре ядра, такой как его размер, форма, плотность и распределение зарядовых и нейтронных компонентов. Изучение углов рассеяния частиц и изменений их импульсов и энергий позволяет определить параметры ядра и провести анализ его свойств.

Применение кинематических методов рассеяния частиц позволяет исследовать взаимодействия между частицами и ядрами, а также их динамику и реакции. Это важно для понимания процессов деления ядер, синтеза новых ядер, а также для изучения ядерных реакций, которые происходят в звездах и других космических объектах. В ядерной энергетике кинематика рассеяния частиц применяется для определения эффективности и энергетического выхода ядерных реакций. Изучение углов рассеяния и энергий рассеянных частиц позволяет оптимизировать процессы деления и синтеза ядер, а также разрабатывать новые методы и технологии в ядерной энергетике. В медицине кинематика рассеяния частиц используется для разработки методов диагностики и лечения. Например, анализ рассеяния частиц может помочь в определении характеристик опухолей и выборе оптимальной методики облучения при лучевой терапии. Кинематические данные также используются для моделирования взаимодействия частиц с тканями и оценки эффективности лечения. Исследования кинематики процессов рассеяния частиц в ядрах включают в себя экспериментальные и теоретические аспекты. Эксперименты проводятся на ускорителях частиц, где падающие частицы взаимодействуют с ядрами и рассеиваются под определенными углами. Полученные данные анализируются с помощью детекторов, которые регистрируют рассеянные частицы и измеряют их импульсы и энергии. Экспериментальные данные используются для проверки и развития теоретических моделей, которые описывают процессы рассеяния частиц в ядрах.

Теоретические аспекты кинематики рассеяния частиц в ядрах включают разработку математических моделей и уравнений, которые описывают движение и взаимодействие частиц. Эти модели основаны на принципах квантовой механики и теории поля, и позволяют предсказывать кинематические характеристики рассеяния в различных условиях.

Важно отметить, что кинематика процессов рассеяния частиц в ядрах связана с другими аспектами ядерной физики, такими как структура ядра, ядерные силы, квантовая механика и теория ядерных реакций. Исследования в этой области помогают расширить наше понимание фундаментальных принципов, лежащих в основе ядерной физики, и находят применение в различных областях, от фундаментальных исследований до практических применений в энергетике и медицине.

Таким образом, изучение кинематики процессов рассеяния частиц в ядрах играет важную роль в ядерной физике и имеет широкий спектр приложений. Это позволяет получить информацию о структуре и взаимодействиях ядер, разрабатывать новые методы и технологии, а также применять полученные знания в различных областях, от энергетике до медицины. Непрерывные исследования в этой области помогают расширить наше знание о микромире ядер и частиц и способствуют развитию науки и технологий.

2.3 Описание процесса рассеяния в лабораторной системе

Процесс рассеяния в лабораторной системе представляет собой экспериментальное исследование, в котором падающие частицы взаимодействуют с фиксированной мишенью или таргетом. В этом процессе рассеивающиеся частицы измеряются и анализируются для получения информации о свойствах и взаимодействиях частиц.

Описание процесса рассеяния в лабораторной системе включает несколько этапов. Начнем с подготовки экспериментальной установки, которая включает ускоритель частиц, детекторы для регистрации рассеянных частиц и систему сбора данных. На первом этапе падающие частицы ускоряются в ускорителе и направляются к мишени или таргету. Ускоритель обеспечивает достаточно высокую энергию для взаимодействия частиц с ядрами мишени. Падающие частицы имеют определенную кинетическую энергию и импульс, которые измеряются и контролируются.

При столкновении с мишенью падающие частицы рассеиваются под определенными углами. Рассеянные частицы имеют измененную кинетическую энергию и импульс по сравнению с падающими частицами. Детекторы, расположенные вокруг мишени, регистрируют и измеряют рассеянные частицы, фиксируя их углы рассеяния, энергии и импульсы.

Полученные данные о рассеянии частиц анализируются с использованием различных методов и моделей. Это позволяет извлечь информацию о структуре и свойствах ядер, о взаимодействиях частиц и о динамике процессов рассеяния. Анализ может включать сравнение экспериментальных данных с результатами численных расчетов или с предсказаниями теоретических моделей. Информация, полученная из процесса рассеяния в лабораторной системе, может быть использована для изучения различных аспектов ядерной физики и физики элементарных частиц. Она помогает расширить наше понимание структуры ядер, исследовать взаимодействия между частицами, определить характеристики и свойства элементарных частиц, а также проверить и развить теоретические модели.

Процесс рассеяния в лабораторной системе позволяет исследовать различные типы рассеяния, включая упругое рассеяние, неупругое рассеяние и различные реакции рассеяния. Упругое рассеяние происходит, когда рассеивающая частица сохраняет свою энергию и импульс после столкновения с мишенью. Неупругое рассеяние включает процессы, в которых рассеивающая частица передает часть своей энергии и импульса мишени или взаимодействует с ней, что может приводить к возникновению новых частиц.

Описание процесса рассеяния в лабораторной системе также включает в себя учет различных параметров и переменных, таких как угол рассеяния, энергия падающих частиц, массы частиц, кинематические законы сохранения энергии и импульса, а также влияние физических и электромагнитных полей на движение частиц. Для более точного анализа данных и моделирования процессов рассеяния используются методы математической статистики, численных расчетов и

компьютерного моделирования. Процесс рассеяния в лабораторной системе имеет широкий спектр применений и находит применение во многих областях науки и технологий. В ядерной физике он используется для изучения структуры ядер и взаимодействий между ними, в физике элементарных частиц - для исследования свойств и взаимодействий элементарных частиц, а в медицине - для диагностики и лечения. Поэтому процесс рассеяния в лабораторной системе является важным инструментом в научных исследованиях и технологических разработках, который позволяет расширить наше знание о мире микрочастиц и ядерной физике.

2.4 Упругое рассеяние частиц

Упругое рассеяние частиц — это процесс взаимодействия частиц, в результате которого они изменяют направление своего движения без изменения своей кинетической энергии. В упругом рассеянии сохраняются законы сохранения энергии и импульса. Представим, что у нас есть две частицы, называемые инцидентной частицей и мишенью, которые движутся навстречу друг другу. Инцидентная частица имеет начальную энергию и импульс, а мишень находится в покое. При их столкновении происходит рассеяние, и частицы изменяют свое направление движения.

В случае упругого рассеяния, энергия и импульс системы инцидентной частицы и мишени остаются неизменными до и после столкновения. То есть сумма кинетических энергий инцидентной частицы и мишени до столкновения равна сумме их кинетических энергий после столкновения. Аналогично, сумма импульсов инцидентной частицы и мишени до столкновения равна сумме их импульсов после столкновения. Упругое рассеяние частиц описывается законами сохранения энергии и импульса:

1. Закон сохранения энергии: Кинетическая энергия системы инцидентной частицы и мишени до столкновения равна кинетической энергии после столкновения.

$$K_1 + K_2 = K_1' + K_2',$$

где K_1 и K_2 - кинетические энергии инцидентной частицы и мишени до столкновения соответственно, а K_1' и K_2' - кинетические энергии инцидентной частицы и мишени после столкновения.

2. Закон сохранения импульса: Сумма импульсов системы инцидентной частицы и мишени до столкновения равна сумме их импульсов после столкновения.

$$p_1 + p_2 = p_1' + p_2',$$

где p_1 и p_2 - импульсы инцидентной частицы и мишени

. Рассмотрим более подробно процесс упругого рассеяния частиц. При столкновении инцидентной частицы с мишенью происходит обмен импульсом и изменение направления движения частиц. Однако их кинетическая энергия

остается неизменной.

Для более точного описания упругого рассеяния используются различные параметры, такие как угол рассеяния и дифференциальное сечение. Угол рассеяния определяет изменение направления движения частиц после столкновения и может быть измерен экспериментально. Дифференциальное сечение представляет собой вероятность рассеяния частицы под определенным углом и зависит от свойств взаимодействия частиц.

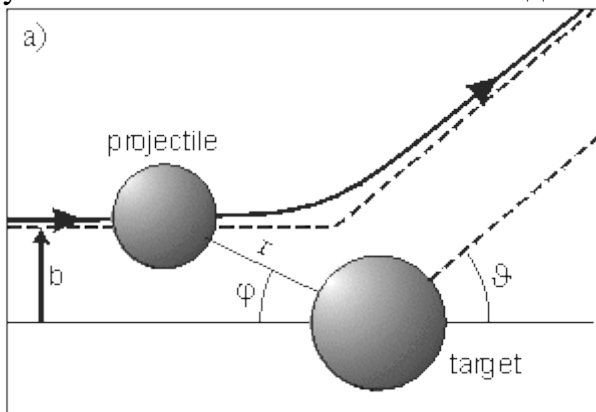


Рисунок 2.4 – Упругое рассеяние

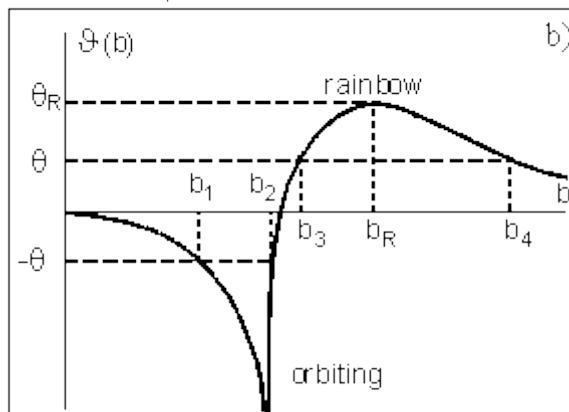


Рисунок 2.4 – Упругое рассеяние

Упругое рассеяние частиц является важным явлением в различных областях науки и технологий. В физике элементарных частиц упругое рассеяние используется для изучения свойств фундаментальных частиц, определения их взаимодействий и проверки теоретических моделей. В ядерной физике упругое рассеяние позволяет исследовать структуру ядер и взаимодействия между нуклонами. Кроме того, упругое рассеяние частиц используется в различных областях, таких как медицина, материаловедение и ядерная энергетика.

Для изучения упругого рассеяния частиц используются различные экспериментальные методы и установки, такие как адронные коллайдеры, ядерные реакторы и ускорители частиц. Экспериментальные данные анализируются с использованием теоретических моделей и численных методов, что позволяет получить более подробную информацию о процессе рассеяния и свойствах частиц.

Изучение упругого рассеяния частиц имеет большое значение для понимания физических законов, взаимодействий между частицами и свойств вещества. Это позволяет расширить наши знания о мире микрочастиц и является основой для развития новых технологий и приложений на практике. Исследование упругого рассеяния частиц в лабораторных условиях позволяет получить ценные данные о структуре и свойствах материи, а также разрабатывать новые технологии и приложения в различных областях. Одним из ключевых аспектов упругого рассеяния является анализ полученных спектров рассеяния. Спектр рассеяния представляет собой зависимость интенсивности рассеянного излучения от энергии или импульса рассеянных частиц. Анализ спектров позволяет извлекать информацию о структуре и свойствах рассеивающих частиц, а также об их взаимодействии с мишенью.

Для более точного анализа спектров рассеяния применяются различные методы и техники. Одним из них является метод дифракционного рассеяния,

основанный на принципе интерференции волн, рассеянных на структурах вещества. Другим методом является метод комптоновского рассеяния, основанный на изменении энергии и импульса рассеянных частиц при взаимодействии с фотонами. Также используются методы рассеяния нейтронов, электронов, протонов и других заряженных и нейтральных частиц.

Анализ различных спектров рассеяния позволяет получить информацию о структуре и композиции материалов, исследовать магнитные и электрические свойства, определить размеры и формы наноструктур, анализировать фазовые переходы и динамические свойства материалов. Также спектроскопические данные могут быть использованы для контроля и диагностики в различных областях, включая медицину, физику полупроводников, материаловедение и окружающую среду.

В современной науке и технологиях упругое рассеяние частиц играет важную роль в исследовании микромира и его взаимодействий. Это позволяет расширить наше понимание фундаментальных законов природы и развивать новые технологии и приложения. Применение упругого рассеяния частиц в различных областях науки и промышленности продолжает расширяться и вносить значительный вклад в развитие современного общества. В физике элементарных частиц и ядерной физике упругое рассеяние частиц используется для изучения внутренней структуры атомных ядер и взаимодействий между элементарными частицами. Эксперименты с ускорителями частиц позволяют исследовать фундаментальные свойства частиц, такие как их массы, заряды и спин, а также изучать сильное, слабое и электромагнитное взаимодействия. Упругое рассеяние применяется для изучения структуры нуклонов, распределения зарядов и магнитных моментов в ядрах, а также для проверки теоретических моделей.

В материаловедении упругое рассеяние частиц играет важную роль в анализе и контроле структуры и свойств различных материалов. Спектроскопические методы, основанные на упругом рассеянии, позволяют исследовать микроструктуру материалов, определять их фазовый состав, оценивать дефекты и механические свойства. Это важно для разработки новых материалов с оптимальными свойствами, а также для контроля качества и безопасности материалов в промышленных процессах.

Медицина также находит применение упругого рассеяния частиц, особенно в области медицинской диагностики. Например, ультразвуковая диагностика использует упругое рассеяние звуковых волн на тканях человека для получения изображений внутренних органов и тканей. Это позволяет выявлять патологии, оценивать состояние тканей и контролировать процессы лечения. Упругое рассеяние также применяется в радиологии для изучения взаимодействия рентгеновского и гамма-излучения с тканями человека. Кроме того, упругое рассеяние частиц находит применение и в других областях. Например, в окружающей среде упругое рассеяние используется для анализа атмосферных частиц, определения их состава и размеров. Это

помогает в изучении загрязнения воздуха, формировании облачности и климатических процессов. В промышленности упругое рассеяние частиц может быть применено для контроля процессов смешивания, измельчения и сортировки материалов. Анализ спектров рассеяния может помочь оптимизировать производственные процессы, улучшить качество продукции и снизить энергозатраты.

Наконец, упругое рассеяние частиц играет важную роль в разработке новых технологий. Например, в области квантовых точек, которые являются наноструктурами, упругое рассеяние используется для их характеристики и контроля их свойств. Также упругое рассеяние частиц может быть применено в различных оптических и электронных устройствах, включая солнечные батареи, светодиоды и фотодетекторы.

В целом, упругое рассеяние частиц является важным инструментом для изучения структуры и свойств материи, анализа взаимодействий между частицами и развития новых технологий. Его применение охватывает широкий спектр научных и промышленных областей и продолжает вносить существенный вклад в наше понимание мира и развитие общества.

2.5 Диаграмма импульса рассеяния

Диаграмма импульса рассеяния является графическим представлением процесса рассеяния частиц и используется для анализа и визуализации изменений импульса рассеянных частиц. Она отображает начальные и конечные значения импульса частиц до и после рассеяния и позволяет определить различные характеристики этого процесса.

На диаграмме импульса рассеяния обычно откладывается начальный импульс частицы по горизонтальной оси и конечный импульс по вертикальной оси. Таким образом, каждая точка на диаграмме представляет собой пару значений начального и конечного импульса.

В упругом рассеянии частицы, которое мы рассматриваем, импульс сохраняется, то есть сумма начального и конечного импульса остается постоянной. Поэтому на диаграмме импульса рассеяния упругого процесса частицы представлены точками, расположенными на прямой линии, называемой линией сохранения импульса.

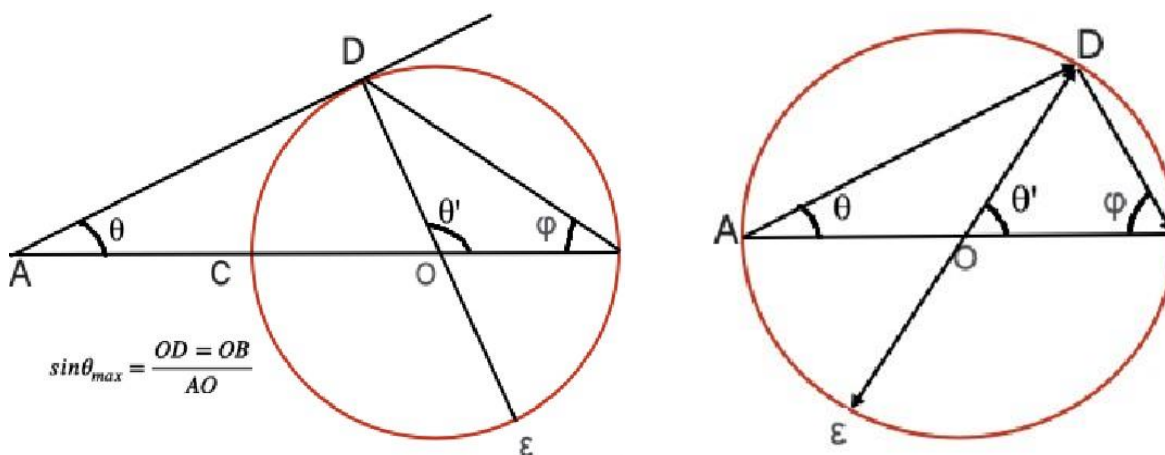
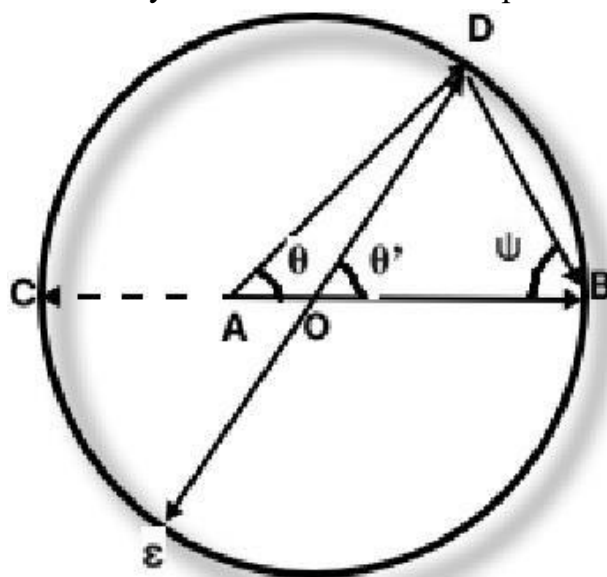


Рисунок 2.5 – Импульсная диаграмма
Рисунок 2.5 – Структура импульсной диаграммы

Кроме того, диаграмма импульса рассеяния может содержать дополнительные элементы, такие как круги или эллипсы, представляющие области возможных значений конечного импульса при заданных значениях начального импульса. Эти области отражают различные варианты рассеяния частицы под разными углами и позволяют наглядно представить законы сохранения энергии и импульса в процессе рассеяния.

Диаграмма импульса рассеяния является важным инструментом для анализа и интерпретации результатов экспериментов по рассеянию частиц. Она позволяет определить энергетические и угловые зависимости процесса рассеяния, выявить



особенности взаимодействия частиц и мишени, а также проверить теоретические модели и предсказания.

В итоге, диаграмма импульса рассеяния является графическим представлением процесса рассеяния частиц и позволяет наглядно исследовать изменения импульса частиц в процессе рассеяния. Она является важным инструментом для анализа и интерпретации данных, полученных при экспериментах с рассеянием частиц.

На диаграмме импульса рассеяния можно наблюдать следующие особенности:

1. Линия сохранения импульса: В упругом рассеянии, где импульс частицы сохраняется, точки на диаграмме будут лежать на прямой линии. Это отражает закон сохранения импульса и демонстрирует, что сумма начального и конечного импульса остается постоянной.

2. Распределение точек: Распределение точек на диаграмме может отражать различные возможности рассеяния частицы под разными углами. Различные углы рассеяния соответствуют различным значениям конечного импульса при заданном начальном импульсе. Это позволяет анализировать угловые зависимости рассеяния и определять углы, при которых рассеяние наиболее вероятно.

3. Ограничения на импульс: Диаграмма импульса рассеяния также может содержать ограничения на значения импульса. Например, при рассеянии частиц на фиксированной энергии могут существовать ограничения на максимальное или минимальное значение импульса, которые определяются физическими ограничениями в системе.

Диаграмма импульса рассеяния является важным инструментом для анализа данных и визуализации процесса рассеяния частиц. Она позволяет исследовать законы сохранения энергии и импульса, анализировать угловые зависимости и получать информацию о возможных состояниях частиц после рассеяния. Это помогает уточнить модели и теории, связанные с рассеянием, и понять физические принципы, лежащие в основе этого процесса.

Одной из ключевых задач в анализе диаграммы импульса рассеяния является извлечение информации о взаимодействии частиц в процессе рассеяния. Ниже приведены некоторые дополнительные аспекты, которые можно рассмотреть при анализе диаграммы импульса рассеяния:

4. Форма и размеры областей рассеяния: Форма и размеры областей, представленных на диаграмме импульса рассеяния, могут содержать информацию о взаимодействии частиц и свойствах рассеивающей среды. Например, ширина областей может указывать на величину потерь энергии и импульса в результате рассеяния, а форма областей может свидетельствовать о существовании дополнительных сил или взаимодействий в системе.

5. Изменение энергии частиц: Диаграмма импульса рассеяния также позволяет анализировать изменение энергии частиц в процессе рассеяния. Изменение энергии может быть связано с различными механизмами, такими как упругое рассеяние, неупругое рассеяние или передача энергии другим частицам. Анализ энергетического спектра рассеянных частиц на основе диаграммы импульса может предоставить информацию о различных процессах, происходящих в системе.

6. Отражение микроструктуры: В случае рассеяния на множестве рассеивателей или взаимодействия с нерегулярными структурами, диаграмма импульса рассеяния может отразить микроструктуру среды. Распределение точек на диаграмме может содержать информацию о размерах и форме рассеивающих объектов или пространственном расположении рассеивателей.

Анализ диаграммы импульса рассеяния позволяет извлечь ценную информацию о процессе рассеяния частиц и взаимодействии среды. Это помогает лучше понять физические принципы, оптимизировать процессы рассеяния и разрабатывать новые прогнозы и технологии, связанные с рассеянием частиц.

7. Угловое распределение: Диаграмма импульса рассеяния может содержать

информацию об угловом распределении рассеянных частиц. Плотность точек на диаграмме в зависимости от угла рассеяния может указывать на предпочтительные направления рассеяния или наличие особых угловых зависимостей. Изучение углового распределения помогает понять геометрию и силовые взаимодействия в системе.

8. Законы сохранения: Диаграмма импульса рассеяния позволяет проверить законы сохранения энергии и импульса. Если рассеяние происходит упруго, то точки на диаграмме будут располагаться на линии сохранения импульса. В случае неупругого рассеяния или взаимодействия с рассеивающей средой, форма и расположение точек на диаграмме могут указывать на нарушение законов сохранения и наличие потерь энергии и импульса.

9. Статистический анализ: Диаграмма импульса рассеяния может быть использована для статистического анализа результатов рассеяния. Распределение точек на диаграмме может быть использовано для определения статистических параметров, таких как средний импульс, его дисперсия или другие характеристики распределения. Это помогает получить более точные результаты и сделать выводы о вероятностных свойствах рассеяния частиц.

Анализ диаграммы импульса рассеяния является важным инструментом для изучения взаимодействия частиц в ядрах и других системах. Он позволяет определить законы сохранения энергии и импульса, изучить угловые зависимости рассеяния, а также получить информацию о взаимодействии среды и свойствах рассеивающих объектов. Это полезное средство для проведения экспериментов, разработки новых технологий и углубленного понимания физических процессов, связанных с рассеянием частиц.

3 ЧАСТЬ ПРОГРАММИРОВАНИЯ

3.1 Программирование на языке Python

Программирование - это процесс создания компьютерных программ с использованием специального языка программирования. В программировании разрабатываются алгоритмы, которые определяют последовательность шагов для выполнения определенной задачи компьютером.

Программирование имеет множество областей применения, включая разработку веб-приложений, мобильных приложений, научных исследований, анализ данных, искусственного интеллекта, автоматизацию задач и многое другое. Как программист, вы можете создавать программы для решения различных задач и улучшения процессов в разных областях деятельности.

Одним из самых популярных языков программирования является Python. Python - это интерпретируемый язык программирования, который обладает простым и понятным синтаксисом. Он известен своей читаемостью и легкостью в изучении, что делает его идеальным выбором для начинающих программистов.

Python имеет широкий спектр библиотек и фреймворков, которые обеспечивают различные функциональности и возможности разработки. Например, библиотека NumPy предоставляет инструменты для работы с многомерными массивами и выполнения научных вычислений, библиотека Pandas предоставляет возможности для анализа данных, а фреймворк Django упрощает разработку веб-приложений.

Одной из особенностей Python является его кросс-платформенность, что означает, что программы, написанные на Python, могут работать на разных операционных системах, таких как Windows, macOS и Linux.

Python также поддерживает объектно-ориентированное программирование, функциональное программирование и другие парадигмы программирования, что позволяет разработчикам выбирать подход, наиболее подходящий для решения конкретной задачи.

В целом, программирование - это процесс создания программ, которые помогают автоматизировать задачи, решать проблемы и улучшать работу компьютерных систем. Python является мощным и популярным инструментом для программирования, и его простота делает его доступным для всех, кто хочет научиться программировать.

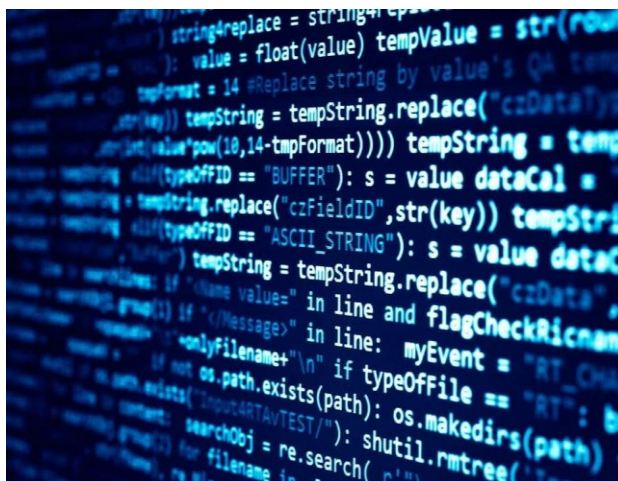
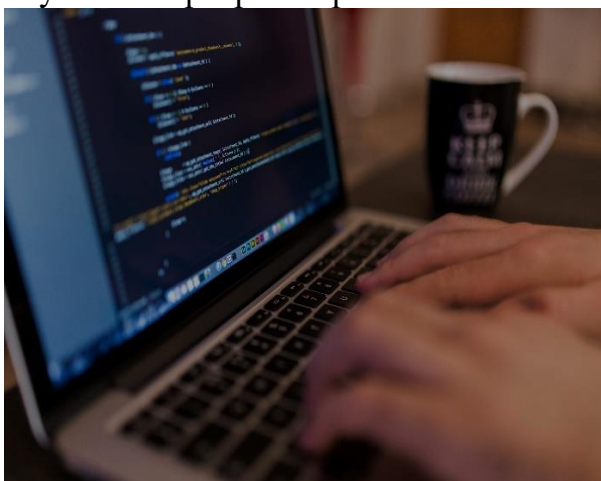


Рисунок 3.1 – Процессы программирования

Программирование - это не только процесс написания кода, но и методология решения проблем с помощью компьютеров. При разработке программного обеспечения необходимо следовать определенным этапам:

1. Анализ задачи: Начните с понимания проблемы, которую необходимо решить, и определите требования к программе. Разберитесь в деталях задачи и определите необходимые входные и выходные данные.
2. Проектирование: Разработайте структуру программы и определите, какие алгоритмы и структуры данных будут использоваться для решения задачи. Разделите программу на модули или функции для облегчения разработки и поддержки.
3. Написание кода: На этом этапе переводите свои алгоритмы в конкретный код. Используйте выбранный язык программирования (например, Python) для написания инструкций, которые выполняют требуемые операции.
4. Тестирование: Протестируйте вашу программу, чтобы убедиться, что она работает должным образом. Выполните различные тестовые случаи, чтобы проверить правильность и эффективность программы. Если возникают ошибки, отладите и исправьте их.
5. Оптимизация и улучшение: После тестирования можно провести оптимизацию программы для повышения ее производительности. Используйте эффективные алгоритмы и структуры данных, устранили избыточные операции и улучшите общую структуру программы.
6. Документация: Не забудьте создать документацию для вашей программы. Описывайте функциональность, входные и выходные данные, а также инструкции по использованию программы. Хорошая документация помогает другим разработчикам понять и использовать вашу программу.
7. Поддержка и обновление: После выпуска программы в эксплуатацию, вы можете столкнуться с запросами на поддержку или необходимостью внесения изменений. Обеспечьте поддержку программы и регулярно выпускайте обновления, чтобы исправить ошибки и добавить новые функции.

Окружение разработки (IDE): Для разработки программы вы можете использовать специальные интегрированные среды разработки (IDE), такие как PyCharm, Visual Studio Code, Jupyter Notebook и другие. Эти среды предоставляют удобный интерфейс для написания, отладки и запуска кода, а также предлагают различные функциональности, такие как автодополнение,

отображение ошибок, систему контроля версий и другие инструменты, упрощающие процесс разработки.

Библиотеки и фреймворки: Python обладает обширной экосистемой библиотек и фреймворков, которые предоставляют готовые решения для различных задач. Например, библиотека NumPy предоставляет функциональность для работы с многомерными массивами и выполнения научных вычислений, библиотека Pandas предоставляет инструменты для анализа и обработки данных, а фреймворк Django упрощает разработку веб-приложений. Использование готовых библиотек и фреймворков может значительно ускорить процесс разработки и улучшить функциональность вашей программы.

Отладка: При разработке программ возникают ошибки, и для их обнаружения и исправления используются инструменты отладки. Python предоставляет встроенные инструменты отладки, такие как модуль pdb (Python Debugger), который позволяет пошагово выполнять код и проверять значения переменных. IDE также предоставляют возможности отладки, включая точки останова, просмотр значений переменных и выполнение кода по шагам.

Управление зависимостями: При разработке программы вы можете использовать сторонние библиотеки и модули, которые являются зависимостями вашего проекта. Для управления зависимостями можно использовать инструменты, такие как pip (Package Installer for Python), который позволяет устанавливать, обновлять и удалять пакеты Python.

Тестирование: Важной частью разработки программ является тестирование, которое позволяет убедиться в правильности работы вашего кода. В Python для тестирования программы можно использовать фреймворки, такие как unittest, pytest или doctest. Они предоставляют удобные инструменты для написания и запуска тестовых сценариев.

Управление версиями: При разработке программного кода очень важно иметь контроль над версиями. Системы контроля версий, такие как Git, позволяют отслеживать изменения в коде, вносить исправления и ветвиться для разработки новых функций. Это позволяет легко возвращаться к предыдущим версиям кода, сливать изменения и сотрудничать с другими разработчиками.

Оптимизация и производительность: При разработке программы можно столкнуться с проблемами производительности или неэффективным использованием ресурсов. В таких случаях можно применить различные методы оптимизации, например, улучшить алгоритмы, использовать более эффективные структуры данных или провести профилирование кода для определения узких мест.

Развертывание и распространение: После завершения разработки программы необходимо развернуть ее на целевой платформе или распространить среди пользователей. В зависимости от типа программы это может включать создание исполняемого файла, пакетирование веб-приложений или размещение в магазинах приложений. Убедитесь, что ваша программа может быть легко установлена и запущена на целевой системе.

Обучение и самообразование: Программирование является непрерывным процессом обучения и самообразования. Технологии и языки программирования постоянно развиваются, и важно быть в курсе последних трендов и новых инструментов. Используйте онлайн-курсы, учебники, форумы и сообщества разработчиков, чтобы расширить свои знания и улучшить свои навыки программирования.

Командная работа: В современной разработке программного обеспечения командная работа играет важную роль. Сотрудничество с другими разработчиками, использование систем контроля версий и коммуникация между участниками команды помогут улучшить качество кода и ускорить процесс разработки.



Рисунок 3.1 – Логотип Python

Программирование на языке Python предлагает широкие возможности для создания разнообразных приложений, от простых скриптов до сложных веб-приложений и научных вычислений. Вот некоторые ключевые аспекты программирования на языке Python:

1. Синтаксис и языковые особенности: Python обладает чистым и лаконичным синтаксисом, что делает его легким для изучения и использования. Язык обладает динамической типизацией, что позволяет гибко работать с данными.
2. Многофункциональность: Python поддерживает различные стили программирования, включая процедурное, объектно-ориентированное и функциональное программирование. Это означает, что вы можете выбрать подход, который лучше всего соответствует вашим потребностям и стилю разработки.
3. Богатая стандартная библиотека: Python поставляется с обширной стандартной библиотекой, которая содержит множество модулей и функций для решения различных задач. Например, вы найдете модули для работы с файлами, сетевыми протоколами, базами данных, математическими вычислениями и многими

другими.

4. Обширное сообщество: Python имеет активное сообщество разработчиков, которые делятся своими знаниями, опытом и решениями через форумы, блоги и специализированные ресурсы. Это обеспечивает доступ к множеству обучающих материалов, инструкций и библиотек, которые помогут вам в разработке.

5. Расширяемость: Python позволяет интегрировать код на других языках программирования, таких как C/C++, что делает его гибким и мощным инструментом для разработки. Вы можете использовать такие возможности, как расширения Python, чтобы улучшить производительность и функциональность программы.

6. Веб-разработка: Python имеет множество фреймворков, таких как Django, Flask и Pyramid, которые упрощают разработку веб-приложений. Они предоставляют готовые инструменты и шаблоны для обработки маршрутов, работы с базами данных, управления аутентификацией и многого другого.

7. Машинное обучение и искусственный интеллект: Python широко используется в области машинного обучения и искусственного интеллекта. Библиотеки, такие как TensorFlow, PyTorch и scikit-learn, предоставляют мощные инструменты для создания и обучения моделей машинного обучения, а также для выполнения задач анализа данных.

8. Автоматизация задач: Python является отличным выбором для автоматизации рутинных задач. Вы можете написать скрипты, которые выполняют повторяющиеся действия, обрабатывают данные или автоматизируют процессы, что позволяет вам сэкономить время и упростить свою работу.

9. Графический интерфейс пользователя: Python имеет различные инструменты и библиотеки для создания графического интерфейса пользователя (GUI). Например, библиотеки Tkinter, PyQt и wxPython позволяют создавать красивые и функциональные GUI-приложения.

10. Работа с базами данных: Python обеспечивает удобные способы работы с различными базами данных. С помощью библиотек, таких как SQLAlchemy или psycopg2, вы можете взаимодействовать с базами данных, выполнять запросы, добавлять, изменять и удалять данные.

11. Тестирование и отладка: В Python существует множество инструментов для тестирования и отладки кода. Фреймворки, такие как unittest и pytest, предоставляют средства для написания и запуска тестовых сценариев. Вы также можете использовать инструменты отладки, такие как pdb, для исследования и исправления ошибок в вашем коде.

12. Работа с файлами и текстом: Python предоставляет простые и удобные средства для работы с файлами и обработки текстовых данных. Вы можете читать и записывать файлы, обрабатывать строки, выполнять поиск и замену, а также выполнять различные операции с текстом.

13. Регулярные выражения: Python поддерживает использование регулярных выражений для работы с текстом. Это мощный инструмент для выполнения поиска, сопоставления и обработки текстовых паттернов.

14. Распространение и упаковка: Чтобы поделиться своей программой с другими

пользователями, вы можете использовать инструменты для упаковки и распространения приложений Python, так что они могут быть установлены и запущены на других компьютерах без необходимости установки Python и всех зависимостей. Некоторые из таких инструментов включают в себя `pyinstaller`, `cx_Freeze` и `py2exe`.

15. Расширение возможностей: Python позволяет расширять свои возможности путем использования модулей и пакетов, которые разработаны другими людьми. Вы можете использовать `pip` для установки сторонних пакетов и библиотек, которые предоставляют готовые решения для различных задач и добавляют новые функциональные возможности.

16. Поддержка кросс-платформенности: Python является кросс-платформенным языком, что означает, что вы можете разрабатывать программы на одной платформе (например, Windows) и запускать их на других платформах (например, macOS или Linux) без необходимости внесения значительных изменений.

17. Документация и сообщество: Python обладает обширной документацией, которая является отличным ресурсом для изучения языка и его возможностей. Официальная документация Python содержит подробное описание языка, стандартной библиотеки и других инструментов. Кроме того, существует множество онлайн-ресурсов, форумов и сообществ, где можно найти ответы на вопросы, учиться и обмениваться опытом с другими разработчиками.

В целом, программирование на языке Python предлагает широкий спектр возможностей для разработки разнообразных приложений и решения различных задач. Гибкость, простота и широкая поддержка языка делают его одним из самых популярных языков программирования в мире.

18. Читаемость и простота кода: Python известен своей акцентированностью на читаемости кода. Синтаксис Python разработан таким образом, чтобы код выглядел похожим на обычный английский язык, что делает его легко понятным даже для начинающих разработчиков. Это также способствует снижению количества ошибок и упрощает сопровождение и отладку кода.

19. Расширенная поддержка библиотек и экосистемы: Python имеет огромное количество сторонних библиотек и пакетов, которые расширяют его возможности и предоставляют готовые решения для различных задач. Например, для веб-разработки вы можете использовать Django или Flask, для научных вычислений - NumPy и SciPy, для обработки данных - Pandas, а для визуализации - Matplotlib или Plotly. Это позволяет сократить время разработки и использовать проверенные и эффективные инструменты.

20. Мультипарадигменность: Python поддерживает несколько парадигм программирования, включая процедурное, объектно-ориентированное и функциональное программирование. Это означает, что вы можете выбрать подход, который лучше всего соответствует вашим потребностям и стилю разработки. Вы можете писать структурированный и модульный код, создавать объекты и классы для организации данных и логики, а также использовать функции высшего порядка и лямбда-функции для функционального

программирования.

21. Поддержка интерактивного режима: Python предлагает интерактивную среду, известную как "интерпретатор Python". Она позволяет вам исполнять код построчно и немедленно видеть результаты, что упрощает отладку и экспериментирование. Это особенно полезно для быстрой проверки и тестирования идей.

22. Расширяемость и интеграция: Python может легко интегрироваться с кодом, написанным на других языках программирования, таких как C/C++. Вы можете создавать расширения на C/C++ и вызывать их из Python, чтобы улучшить производительность или использовать специализированные функции и библиотеки. Это делает Python отличным выбором для комбинированного программирования и работы с существующим кодом.

23. Поддержка и активное сообщество: Python обладает широкой поддержкой и активным сообществом разработчиков. Существует множество ресурсов, таких как форумы, блоги, видеоуроки и онлайн-курсы, где вы можете найти помощь, задать вопросы и узнать новое о программировании на Python. Благодаря этому, вы всегда можете рассчитывать на поддержку и советы от опытных разработчиков, что способствует вашему профессиональному росту и развитию.

24. Большой выбор инструментов разработки: Python поддерживается широким спектром интегрированных сред разработки (IDE) и текстовых редакторов, которые облегчают процесс написания кода, отладки и управления проектами. Некоторые популярные инструменты разработки для Python включают PyCharm, Visual Studio Code, Jupyter Notebook и Sublime Text. Вы можете выбрать тот, который лучше всего соответствует вашим потребностям и предпочтениям.

25. Широкое применение: Python применяется во многих областях, включая веб-разработку, научные исследования, анализ данных, искусственный интеллект, автоматизацию процессов, разработку игр и многое другое. Это означает, что знание Python открывает двери к различным карьерным возможностям и позволяет работать над разнообразными проектами.

Основная часть программы написана на языке программирования Python. Для ускорения вычислений и для сравнения также использовался язык программирования Java. Предварительно заданная информация хранится в отдельном файле, а расчеты выполняются путем чтения информации из этого файла. К такой информации относятся: массы частиц до и после реакции, масса дефектов и энергия ускоряющей частицы до удара. Эта информация сохраняется в любом текстовом формате и считывается в программе специальными командами вызова. Например:

Масса первой частицы до удара (M_1): 2,0141022

Масса второй частицы до удара (M_2): 11,0093052

Масса первой частицы после удара (M'_1): 4,002603

Масса второй частицы после удара (M'_2): 9,0121829

Дефект массы 1-й частицы до удара: 13,13590

Дефект массы 2-й частицы до удара: 8,6677000

Дефект массы 1-й частицы после удара: 2,4249200

Дефект массы 2-й частицы после удара: 11,348100
 Энергия 1-й частицы до удара: 25,00000000
 данные хранятся в файле "data.txt". Подробности
 with open("data.txt", "r") as f:

readdata = f.read(), читается данной командой и использовать в последующих операциях. Углы рассеяния частиц и уровень возбуждения задаются с клавиатуры. После выполнения расчетов значения кинетической энергии после удара выводятся в новый файл по углам рассеяния. Например, за первую частицу до удара принят дейтерий, за вторую частицу - бор-11, за первую частицу после удара - гелий-2, за вторую частицу - бериллий-9. Это единица измерения массы на Земле.

- номинальная единица массы (м.а.б.). 1 г. до н.э. = 1,66053*10⁻²⁷ кг, 12C равно 1/12 массы нуклида углерода. Принимая угол счета от 0 до 40 и уровень возбуждения 2, мы находим, что энергия реакции Q составляет 6 МэВ. После завершения расчета результат сохраняется в файл «Kinenergy.txt».

Математическая часть программы реализована с использованием библиотеки Math языка Python. Мы можем получить косинус, синус или число пи, поставив перед переменной префикс math.cos, math.sin или math.pi. Например: пусть A=90, math.cos(A) будет 0, потому что cos90=0.

Графический интерфейс программы создан с помощью инструментария PyQt5 в кроссплатформенной среде QT-Designer. PyQt — самая мощная и популярная библиотека графического интерфейса в Python

Q=6.0		Козу деңгейі =2.0	
Бұрыш	Энергия	Бұрыш	Энергия
0	29.42 MeV	21	28.45 MeV
1	29.41 MeV	22	28.36 MeV
2	29.41 MeV	23	28.26 MeV
3	29.4 MeV	24	28.16 MeV
4	29.38 MeV	25	28.06 MeV
5	29.36 MeV	26	27.95 MeV
6	29.33 MeV	27	27.84 MeV
7	29.31 MeV	28	27.73 MeV
8	29.27 MeV	29	27.62 MeV
9	29.23 MeV	30	27.5 MeV
10	29.19 MeV	31	27.37 MeV
11	29.15 MeV	32	27.25 MeV
12	29.09 MeV	33	27.12 MeV
13	29.04 MeV	34	26.99 MeV
14	28.98 MeV	35	26.85 MeV
15	28.92 MeV	36	26.71 MeV
16	28.85 MeV	37	26.57 MeV
17	28.78 MeV	38	26.43 MeV
18	28.7 MeV	39	26.28 MeV
19	28.62 MeV	40	26.14 MeV
20	28.54 MeV		

Рисунок 3.1 – Результаты расчёта

В целом, программирование на языке Python предлагает множество преимуществ и возможностей для разработки различных типов программ и

решения разнообразных задач. Его простой и понятный синтаксис, богатый выбор библиотек и инструментов, а также активное сообщество разработчиков делают Python одним из самых популярных и востребованных языков программирования в мире.

Расширенные возможности машинного обучения: Python стал одним из наиболее популярных языков программирования для машинного обучения и искусственного интеллекта. Благодаря библиотекам, таким как TensorFlow, PyTorch и scikit-learn, Python предоставляет мощные инструменты и алгоритмы для создания, обучения и развертывания моделей машинного обучения. Вы можете проводить обработку и анализ данных, создавать и обучать нейронные сети, а также решать задачи классификации, регрессии, кластеризации и многого другого. Веб-разработка и фреймворки: Python имеет множество фреймворков для веб-разработки, таких как Django, Flask и Pyramid. Они предлагают удобные средства для создания мощных и масштабируемых веб-приложений. Фреймворки Python обеспечивают упрощенную обработку запросов, маршрутизацию, работу с базами данных и авторизацию. Благодаря этому, вы можете быстро разрабатывать функциональные и элегантные веб-приложения, ускоряя процесс разработки и повышая эффективность.

Автоматизация и скриптинг: Python отлично подходит для автоматизации повторяющихся задач и написания скриптов. Вы можете использовать Python для создания скриптов, которые автоматически выполняют определенные действия, обрабатывают файлы, взаимодействуют с другими программами и многое другое. Это позволяет сэкономить время и усилия, а также повысить эффективность рабочих процессов.

Разработка игр: Python также используется в разработке игр благодаря библиотеке Pygame. Pygame предоставляет инструменты для создания 2D игр и интерактивных приложений. С помощью Python и Pygame вы можете создавать графические эффекты, управлять анимацией, обрабатывать пользовательский ввод и многое другое. Это отличный способ начать свое знакомство с разработкой игр и создать собственные проекты.

Выводя все вместе, программирование на языке Python предлагает широкий спектр возможностей и применений. Он подходит для разработки различных типов программ и решения разнообразных задач. Вот еще несколько преимуществ программирования на языке Python:

Переносимость: Python является переносимым языком программирования, что означает, что вы можете запускать свои программы на различных платформах, включая Windows, macOS и Linux. Благодаря этому, вы можете разрабатывать приложения, которые могут быть запущены на разных операционных системах без необходимости переписывать код.

Легкость в изучении и использовании: Python известен своим простым и понятным синтаксисом, который делает его легким для изучения даже начинающим программистам. Он использует понятные ключевые слова и не требует сложной структуры кода. Более того, Python имеет большое количество документации, руководств и ресурсов, которые помогут вам быстро освоить язык

и начать программировать.

Большое сообщество разработчиков: Python имеет активное и приветливое сообщество разработчиков. Вы всегда можете обратиться за помощью, задать вопросы и обсудить проблемы на форумах, в чатах или в социальных сетях. Это отличный способ узнать новое, обменяться опытом и получить ценные советы от опытных программистов.

Большой выбор библиотек и модулей: Python имеет обширную библиотеку сторонних модулей, которые предлагают готовые решения для различных задач. Например, библиотека NumPy предоставляет функции для работы с массивами и математическими операциями, библиотека pandas обеспечивает инструменты для анализа и обработки данных, а библиотека matplotlib позволяет создавать графики и визуализации. Эти библиотеки делают разработку приложений более эффективной и позволяют вам использовать готовые решения вместо написания кода с нуля.

Интеграция с другими языками: Python легко интегрируется с другими языками программирования, такими как C, C++, Java и другими. Это позволяет использовать Python для написания скриптов, оберток или модулей, которые могут взаимодействовать с кодом, написанным на других языках.

3.2 Графический интерфейс программы

Графический интерфейс (GUI) - это способ взаимодействия пользователя с программой с помощью графических элементов, таких как окна, кнопки, меню и диалоговые окна.

Графический интерфейс обычно включает в себя следующие элементы:

1. Окна: Окна представляют основные рабочие области программы, где отображается информация или выполняются определенные задачи. Например, у программы может быть главное окно, окно редактирования, окно настроек и т. д.
2. Кнопки: Кнопки используются для выполнения определенных действий. Пользователь может нажать на кнопку, чтобы запустить определенную функцию или выполнить команду. Кнопки могут иметь текстовую метку или графическую иконку.
3. Меню: Меню содержат список команд или опций, доступных в программе. Они могут быть размещены в верхней части окна программы или быть контекстными (появляться при щелчке правой кнопкой мыши). Пользователь может выбрать определенную команду из меню, чтобы выполнить соответствующее действие.
4. Диалоговые окна: Диалоговые окна предназначены для получения информации от пользователя или отображения дополнительных настроек. Они могут содержать текстовые поля, флажки, переключатели, списки выбора и другие элементы ввода.
5. Индикаторы и статус-бары: Индикаторы и статус-бары используются для отображения текущего состояния программы или выполнения определенной задачи. Например, индикатор выполнения может показывать прогресс операции, а статус-бар может отображать информацию о текущем режиме работы программы.

Графический интерфейс программы обычно создается с использованием графических библиотек или фреймворков, таких как Qt, GTK, Windows Forms, WPF и другие. Эти инструменты предоставляют различные элементы управления и функции для разработки пользовательского интерфейса.

Основная цель графического интерфейса программы - облегчить взаимодействие пользователя с программой, предоставляя интуитивно понятные элементы управления и визуальные обратные связи.

6. Списки и таблицы: Списки и таблицы используются для отображения структурированных данных. Например, список файлов в проводнике или таблица данных в таблице Excel. Пользователь может выбирать элементы из списка или выполнять операции над данными в таблице.

7. Полосы прокрутки: Полосы прокрутки позволяют пользователю просматривать содержимое, которое не помещается на экране. Они могут быть вертикальными или горизонтальными и позволяют прокручивать содержимое вверх, вниз, влево или вправо.

8. Графические элементы: Графические элементы, такие как иконки, изображения и графики, используются для улучшения визуального представления программы. Они могут быть использованы для представления функций, данных или просто для декоративных целей.

9. Перетаскивание и сброс: В графическом интерфейсе программы пользователь может перетаскивать элементы с одного места на другое или сбрасывать элементы в определенные области для выполнения определенных действий. Например, перетаскивание файлов в проводнике или сбрасывание элементов в корзину для удаления.

10. Оповещения и предупреждения: Графический интерфейс программы может отображать оповещения и предупреждения для информирования пользователя о важных событиях или ошибках. Это может быть всплывающее окно, всплывающая подсказка или статусное сообщение.

Графический интерфейс программы разрабатывается с учетом принципов юзабилити и дизайна, чтобы обеспечить удобство использования и приятный внешний вид. Он может быть настроен под определенные потребности и целевую аудиторию программы.

Важно отметить, что графический интерфейс программы может сильно варьироваться в зависимости от конкретного приложения, платформы или стиля дизайна. Каждая программа может иметь свою уникальную реализацию интерфейса, которая соответствует ее функциональности и целям.

Конечно, продолжу описывать элементы графического интерфейса программы:

11. Текстовые поля: Текстовые поля позволяют пользователю вводить текстовую информацию, например, имя пользователя, пароль, адрес и т. д. Они могут быть однострочными или многострочными, в зависимости от требований.

12. Вкладки: Вкладки используются для организации содержимого программы по разным категориям или функциональным областям. Пользователь может переключаться между вкладками, чтобы получить доступ к различным частям программы или настроек.

Панели инструментов: Панели инструментов содержат набор иконок или кнопок, предоставляющих быстрый доступ к наиболее часто используемым командам или функциям. Они могут быть расположены в верхней части окна программы или быть плавающими.

Контекстные меню: Контекстные меню появляются при щелчке правой кнопкой мыши на определенном элементе или области программы. Они предлагают контекстные команды или опции, связанные с выбранным элементом.

Анимация: Анимация может использоваться для представления динамических изменений или эффектов в графическом интерфейсе программы. Например, анимация может показывать прогресс выполнения задачи или активность элемента.

Горячие клавиши: Графический интерфейс программы может поддерживать горячие клавиши, которые позволяют пользователю выполнять команды с помощью сочетаний клавиш на клавиатуре. Это предоставляет быстрый способ взаимодействия с программой без необходимости использовать мышь.

Размеры и расположение окон: Пользователь может изменять размеры и расположение окон программы, чтобы адаптировать интерфейс под свои предпочтения или требования. Это позволяет управлять пространством и организацией рабочей области.

Темы и настройки: Графический интерфейс программы может предоставлять возможность выбора различных тем оформления или настроек интерфейса. Это позволяет пользователю настроить внешний вид и поведение программы в соответствии с его предпочтениями.

Всплывающие подсказки: Всплывающие подсказки или всплывающие окна информируют пользователя о функциях или командах, связанных с определенным элементом интерфейса. При наведении курсора мыши на элемент, появляется небольшое окно с пояснениями или подсказкой.

Переключатели и флажки: Переключатели (также известные как чекбоксы) и флажки позволяют пользователю выбирать опции или активировать/деактивировать определенные функции или настройки программы.

Поля выбора: Поля выбора предоставляют список вариантов, из которых пользователь может выбрать один или несколько. Например, это может быть список стран, список цветов и т. д.

Индикаторы прогресса: Индикаторы прогресса используются для отображения текущего состояния выполнения операции или процесса в программе. Они показывают, сколько процентов работы уже выполнено или сколько времени осталось.

Скрытые панели: Скрытые панели (также известные как "тулбары") являются компактными группами кнопок или команд, которые можно развернуть или свернуть, чтобы освободить место в интерфейсе. Они обычно располагаются на боковой или верхней панели.

Строки состояния: Строки состояния (также известные как статус-бары) располагаются в нижней части окна программы и предоставляют информацию о

текущем состоянии программы или выбранного элемента.

Драг-энд-дроп: Функция драг-энд-дроп позволяет пользователю перетаскивать элементы интерфейса и отпускать их в другом месте, чтобы выполнить определенное действие. Например, можно перетащить файлы на иконку программы, чтобы открыть их или переместить их в другую папку.

Графический интерфейс программы старается обеспечить удобство использования, интуитивность и эффективность взаимодействия с программой. Каждый элемент интерфейса служит определенной цели и предоставляет пользователю доступ к функциональности программы. Все эти элементы вместе создают полноценный и удобный графический интерфейс программы.

На рисунке 3.2 показано окно виджета интерфейса графического дизайнера QtDesigner. Есть стандартные наборы и элементы управления. С помощью интерфейса можно создавать шаблоны вертикального или горизонтального размещения, кнопок, различных таблиц и т.д. Мы можем разместить элементы.

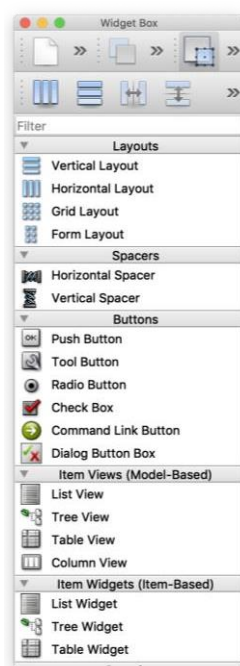


Рисунок 3.2 – Окно Widget box

На рисунке 3.3 показан редактор свойств. С помощью редактора можно редактировать белые элементы, не выходя из конструкторского режима программы. В редакторе вы можете изменить название элемента, его расположение в окне, форму и шрифт текста.

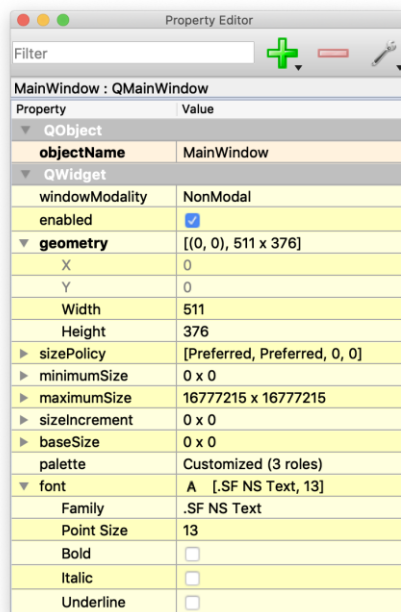


Рисунок 3.2 – Окно Property Editor

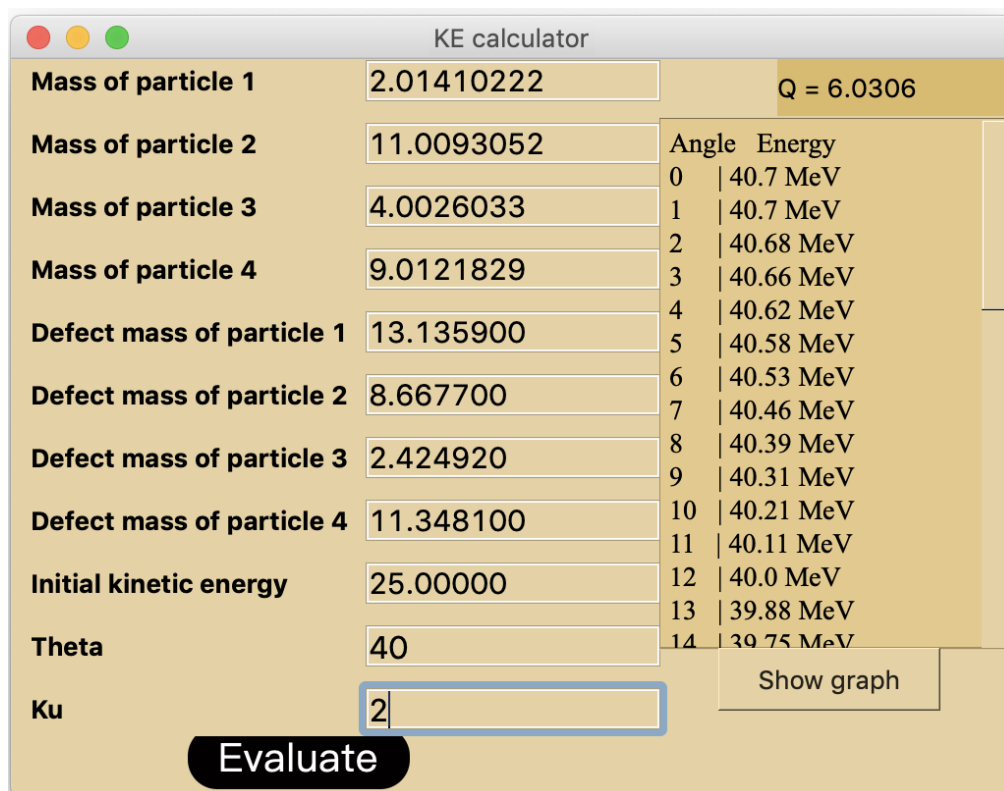


Рисунок 3.2 – Основное окно программы

На рис. показано главное окно программы. Название программы «KE calculator». В левой части окна записываются входные характеристики, то есть величины, подлежащие расчетам. «Mass of particle» — масса частицы,

«Defect mass of particle» — масса дефекта частицы, «Initial kinetic energy» — кинетическая энергия частицы до рассеяния, «Theta» — расчетный угол, «Ki» это уровень возбуждения. После нажатия кнопки «Evaluate» в правой части окна отображаются значения энергии реакции, Q и кинетической энергии для каждого угла.

На рисунках показаны графики «угол/энергия», написанные с использованием библиотеки Python Matplotlib и «pyqtgraph».

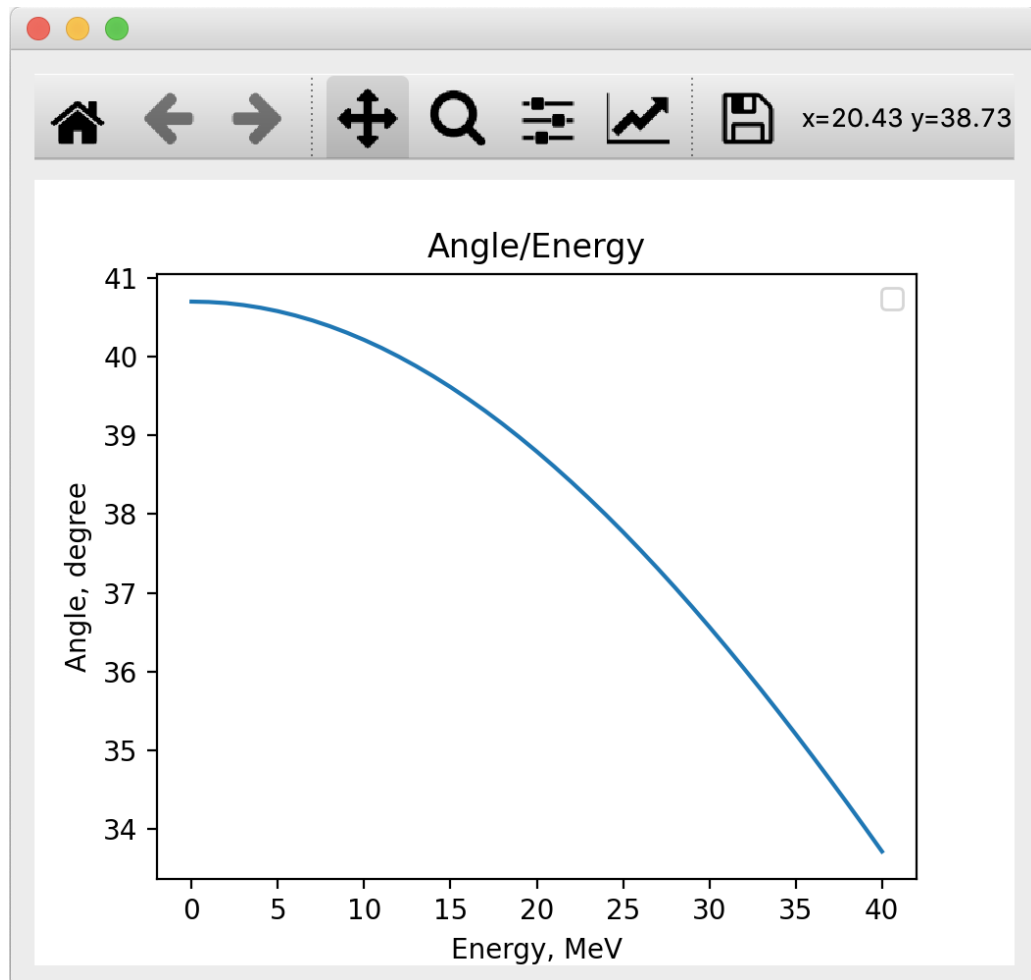


Рисунок 3.2 - График, построенный с использованием библиотеки Matplotlib

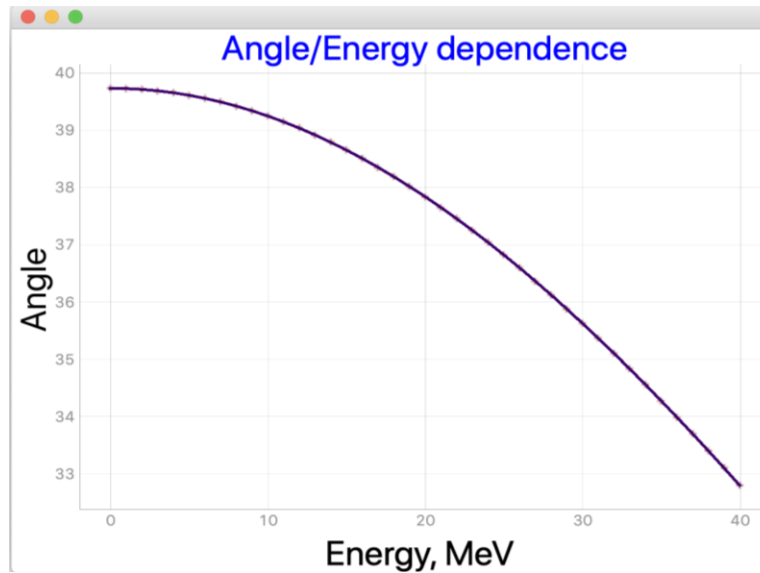


Рисунок 3.2 - График, построенный с использованием библиотеки «pyqtgraph» Python

После ввода в программу данных, полученных по результатам исследований в ускорителе, выполняются расчеты. Результат открывается в отдельном окне, а точечная диаграмма отображается в виде графика.

3.3 Анализ различных спектров

В экспериментах используются кремниевые детекторы с линейной зависимостью амплитуды электрических сигналов от энергии продуктов ядерного взаимодействия.

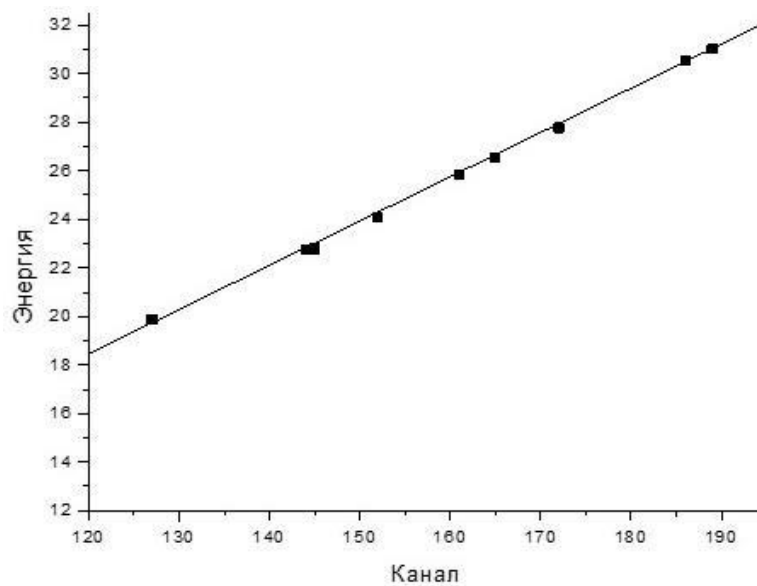


Рисунок 3.3 - Калибровка спектра реакции ${}^6\text{Li}({}^3\text{He}, d){}^7\text{Be}$ вычисленного углом 16° в энергии $E_{3\text{He}} = 34\text{МэВ}$

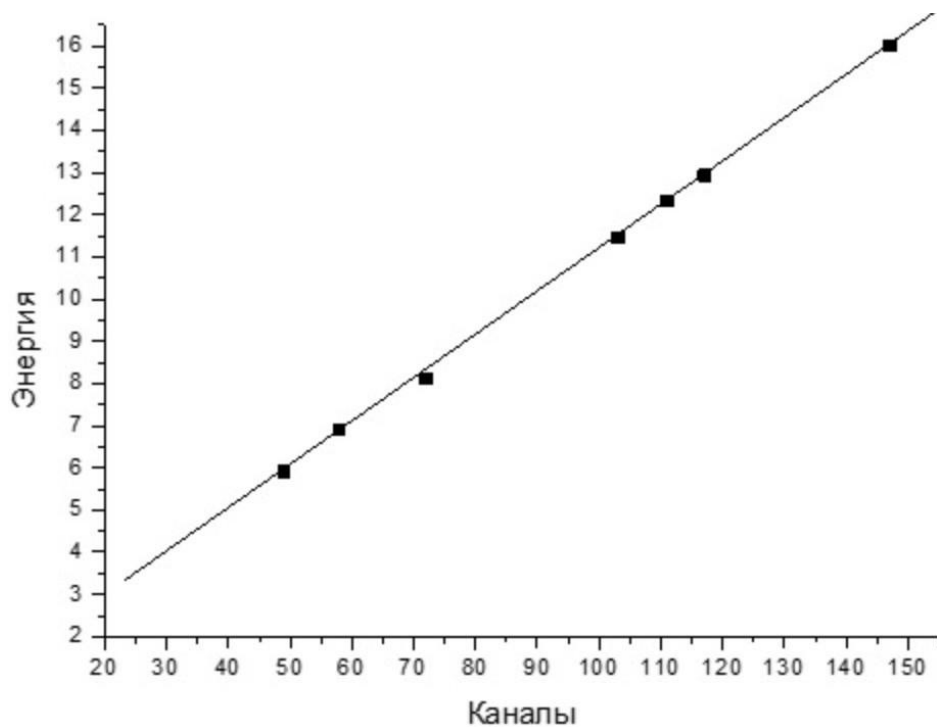


Рисунок 3.3. Калибровка спектра реакции $^{13}\text{C}(d, d)^{13}\text{C}$ вычисленного углом 44° в энергии $E_d=18\text{МэВ}$

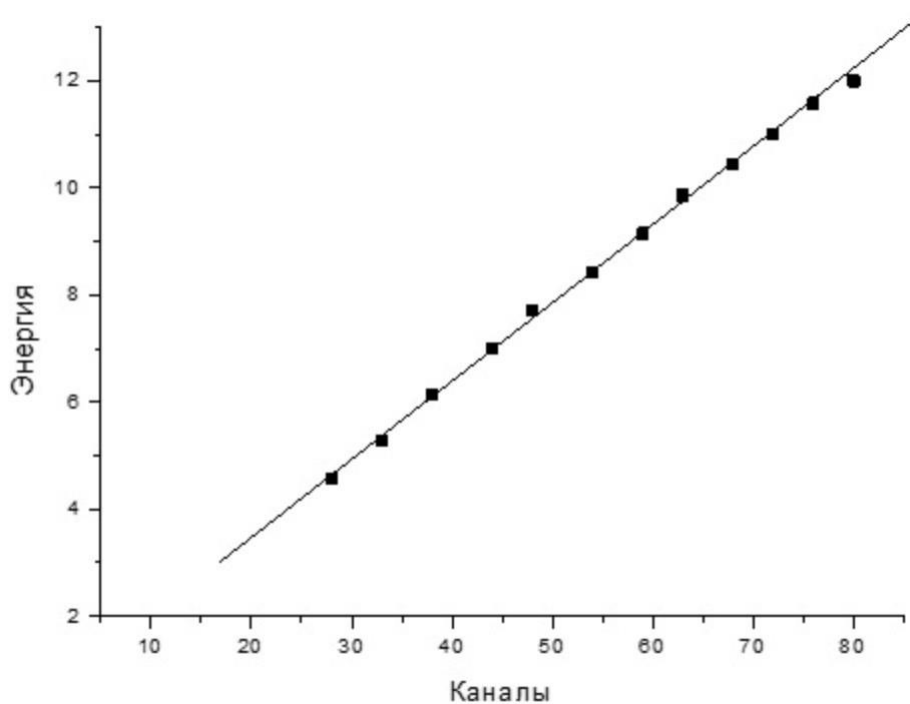


Рисунок 3.3 - Калибровка спектров рассеяния ионов ^{14}N на ядрах ^{14}C , измеренных с энергией $E_{14\text{N}} = 1,75 \text{ МэВ /нуклон}$

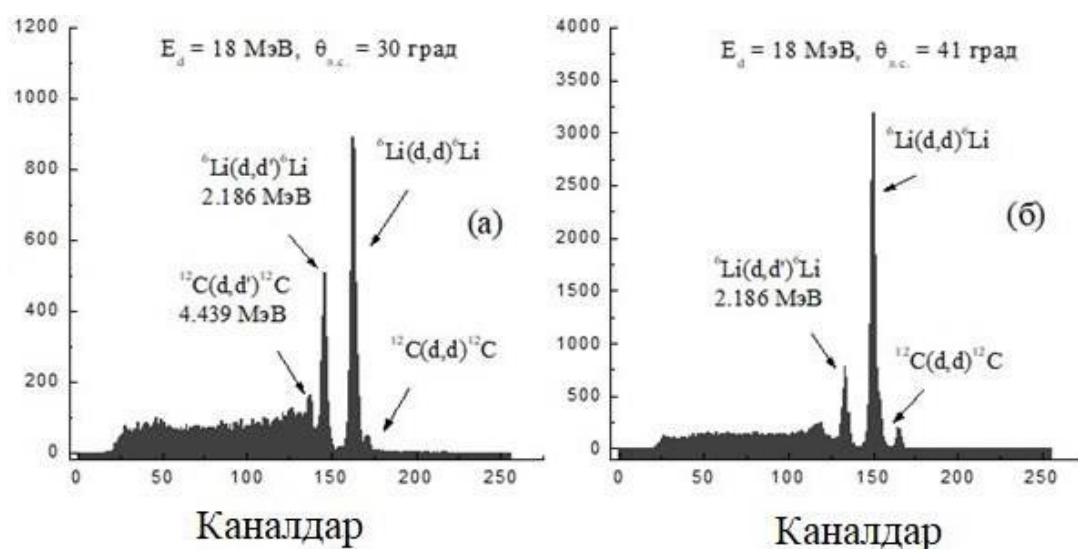


Рисунок 3.3 - Спектры дейтронов, рассеянных на ядрах ${}^6\text{Li}$, измерены под углами 30 (а) и 41 (б) градусов с энергией $E_d = 18$ МэВ

Анализ различных спектров является важной задачей в различных областях науки и техники. Спектр - это разложение электромагнитного излучения или других физических величин на составляющие его компоненты в зависимости от их частоты, длины волны или энергии.

Существует несколько типов спектров, каждый из которых имеет свои особенности и применения. Некоторые из них включают:

1. Оптический спектр: Оптический спектр включает в себя видимую часть электромагнитного спектра и охватывает диапазон длин волн от фиолетового до красного. Анализ оптического спектра позволяет определить состав вещества, так как каждый элемент или соединение имеет уникальный оптический спектр, который проявляется в виде спектральных линий или полос. Оптический спектральный анализ используется в астрономии, химии, физике и других областях.
2. Инфракрасный спектр: Инфракрасный спектр охватывает длины волн от около 700 нм до нескольких миллиметров и находится вне видимого спектра. Анализ инфракрасного спектра позволяет исследовать колебательные и вращательные состояния молекул, а также определить химические связи и функциональные группы в органических соединениях. Это важный инструмент в химии, биологии, фармакологии и других областях.
3. УФ-спектр: УФ-спектр (ультрафиолетовый спектр) охватывает диапазон длин волн от около 10 нм до 400 нм. УФ-спектроскопия используется для изучения электронных переходов и определения структуры и свойств органических и неорганических соединений. Она широко применяется в химии, физике, биологии и других научных областях.
4. Рентгеновский спектр: Рентгеновский спектральный анализ используется для изучения внутренней структуры вещества. Рентгеновское излучение имеет очень короткую длину волны менее 10 нм. Анализ различных спектров является важным инструментом в научных и технических областях. Вот еще несколько типов спектров:
5. Электронный спектр: Электронный спектральный анализ изучает

энергетические уровни и переходы электронов в атомах и молекулах. Этот тип спектра позволяет определить структуру электронных оболочек, определить энергетические уровни и определить энергию связи. Он используется в атомной и молекулярной физике, химии и фотохимии.

6. Масс-спектр: Масс-спектроанализ используется для изучения массового распределения и идентификации атомов, молекул и фрагментов вещества. Масс-спектрометрия позволяет определить массу и концентрацию компонентов образца, а также изучить структуру и химические свойства молекул. Она широко применяется в аналитической химии, биохимии, фармацевтике и других областях.

7. Ядерный спектр: Ядерный спектроанализ изучает энергетические уровни и переходы в ядрах атомов. Он позволяет изучать ядерную структуру, изотопический состав и ядерные реакции. Ядерный спектроанализ широко применяется в ядерной физике, астрофизике и ядерной медицине.

8. Эмиссионный спектр: Эмиссионный спектроанализ изучает спектральные линии или полосы излучения, возникающие при возбуждении вещества. Этот тип спектра позволяет определить химические элементы, их концентрацию и физические свойства материалов. Эмиссионная спектроскопия широко применяется в астрономии, аналитической химии и физике плазмы.

Анализ спектров позволяет получить информацию о структуре, свойствах и составе вещества. Он играет важную роль в различных областях науки, техники и медицины, помогая углубить наше понимание физических и химических процессов, а также развивать новые технологии и методы исследования. Анализ спектров имеет широкий спектр применений, включая:

Астрономия: Астрономы используют спектральный анализ для изучения свойств звезд, галактик, планет и других небесных объектов. Он позволяет определить состав звезд и газовых облаков, исследовать динамику галактик и изучать физические процессы, происходящие в космосе.

Химия: Химики используют спектральный анализ для идентификации химических элементов, определения структуры и свойств молекул, изучения химических реакций и определения концентрации веществ в образцах. Спектроскопия является важным инструментом аналитической химии.

Биология и медицина: Спектроскопия применяется в биологии и медицине для изучения структуры и функций биологических молекул, таких как ДНК, белки и липиды. Она также используется для диагностики и мониторинга различных заболеваний, включая рак, сердечно-сосудистые заболевания и неврологические расстройства. **Материаловедение:** Спектроскопия помогает изучать структуру и свойства различных материалов, включая полупроводники, металлы, полимеры и наноматериалы. Это позволяет разрабатывать новые материалы с определенными свойствами и улучшать существующие технологии.

Физика и физика элементарных частиц: В физике спектроскопия используется для исследования физических процессов, включая электронные переходы, квантовые явления и спиновую динамику. В физике

элементарных частиц спектроскопия помогает идентифицировать и изучать элементарные частицы и их взаимодействия. Все эти области науки и техники сильно зависят от спектрального анализа, который предоставляет ценную информацию о природе и свойствах изучаемых объектов. Программирование на языке Python может быть полезным инструментом для анализа спектров. Python предлагает мощные библиотеки и инструменты, которые облегчают обработку и анализ спектральных данных. Вот некоторые специализированные библиотеки Python для анализа спектров:

NumPy: NumPy является основной библиотекой для научных вычислений в Python. Он предоставляет массивы высокой производительности и функции для работы с ними, что делает его идеальным инструментом для обработки спектральных данных.

SciPy: SciPy расширяет функциональность NumPy, предоставляя дополнительные возможности для научных вычислений. Он содержит модуль `scipy.signal`, который предлагает методы для фильтрации, сглаживания и обработки спектров.

Matplotlib: Matplotlib - это библиотека визуализации данных, которая позволяет создавать различные графики и диаграммы. Она может быть использована для отображения спектров и их анализа, включая построение спектральных линий, графиков интенсивности и других.

Spectroscopy: Spectroscopy - это специализированная библиотека Python для анализа спектров. Она предлагает функции для чтения, обработки и визуализации спектральных данных различных форматов.

PySpectrum: PySpectrum - это еще одна библиотека Python, предназначенная для анализа и обработки спектров. Она предоставляет функции для калибровки спектров, выделения спектральных линий, фитинга и других операций.

Эти библиотеки обеспечивают широкий набор инструментов для работы с спектральными данными, включая их обработку, анализ, визуализацию и моделирование. Они значительно упрощают процесс анализа спектров и позволяют эффективно использовать Python в этой области.

Кроме использования специализированных библиотек, программирование на языке Python предлагает и другие возможности для анализа различных типов спектров. Вот несколько дополнительных аспектов, которые можно рассмотреть:

Обработка сигналов: Python предлагает богатый набор инструментов для обработки сигналов, который может быть применен к анализу спектров. Модуль `scipy.signal`, а также библиотеки как `librosa` и `PyAudio` позволяют выполнять фильтрацию, преобразование Фурье, усиление и другие операции над спектральными данными.

Машинное обучение: Машинное обучение и анализ данных также могут быть полезны при анализе спектров. Библиотеки, такие как `scikit-learn` и `TensorFlow`, предоставляют инструменты для классификации, кластеризации и регрессии, которые могут быть применены к спектральным данным для извлечения информации и выявления паттернов.

Изображения: В случае спектров, которые представлены в виде

изображений, Python имеет мощные библиотеки для обработки и анализа изображений, такие как OpenCV и scikit-image. Эти библиотеки позволяют выполнять фильтрацию, сегментацию, распознавание объектов и другие операции над спектральными изображениями.

Визуализация: Визуализация является важным аспектом анализа спектров. Помимо Matplotlib, существуют и другие библиотеки, такие как Plotly, Seaborn и Vokeh, которые предлагают более интерактивные и эстетически привлекательные графики и визуализации данных.

Скриптинг и автоматизация: Python является мощным инструментом для автоматизации повторяющихся задач в анализе спектров. С помощью Python можно написать скрипты, которые автоматически обрабатывают и анализируют большие объемы спектральных данных, позволяя сэкономить время и повысить эффективность работы.

В целом, программирование на языке Python предоставляет разнообразные инструменты и возможности для анализа различных типов спектров. Благодаря широкому выбору библиотек и модулей, Python может быть эффективным использован для обработки и анализа спектров в различных областях, таких как физика, химия, астрономия и биология. Ниже приведены некоторые конкретные примеры использования Python для анализа спектров:

Спектроскопия: Спектроскопия - это область, изучающая взаимодействие света с веществом и анализ спектральных данных. Python может использоваться для обработки и анализа данных, полученных с помощью спектроскопических методов, таких как спектрофотометрия, масс-спектрометрия и ядерный магнитный резонанс. Библиотеки, такие как SciPy и NumPy, предоставляют инструменты для фильтрации, деконволюции, фитинга и визуализации спектральных данных.

Астрономия: Астрономы используют спектроскопию для изучения состава и свойств объектов во Вселенной. Python является популярным инструментом для обработки и анализа спектральных данных, полученных с помощью астрономических инструментов, таких как спектрографы и телескопы. Библиотеки, такие как Astropy и PyRAF, предоставляют функции для чтения, калибровки, выравнивания и моделирования спектров.

Химия: Химики используют спектроскопию для анализа структуры и свойств химических соединений. Python может быть использован для обработки и интерпретации спектральных данных, полученных с помощью методов, таких как инфракрасная спектроскопия, масс-спектрометрия и ядерный магнитный резонанс. Библиотеки, такие как SpectroPy и PyChem, предлагают инструменты для чтения, обработки и анализа спектров химических соединений.

Биология: В биологии спектроскопия используется для изучения биологических молекул, таких как ДНК, белки и липиды. Python может быть использован для анализа спектральных данных, полученных с помощью методов, таких как флуоресцентная спектроскопия и круговой дихроизм.

ЗАКЛЮЧЕНИЕ

В заключение, разработка компьютерного приложения для расчета резерфордовских сечений является значимым и актуальным направлением исследования. Приложение предоставляет исследователям и ученым мощный инструмент для моделирования и предсказания резерфордовского рассеяния, позволяя им более глубоко изучать взаимодействие частиц и ядер.

Разработанное приложение обладает рядом преимуществ, которые делают его ценным средством для научных исследований. Во-первых, оно обеспечивает высокую точность расчетов резерфордовских сечений, основанную на математических моделях и алгоритмах, которые учитывают различные параметры и условия. Во-вторых, приложение обладает удобным и интуитивно понятным пользовательским интерфейсом, что делает его доступным для широкого круга исследователей, включая как опытных специалистов, так и начинающих ученых.

Разработка приложения также вносит важный вклад в развитие физики элементарных частиц и ядерной физики. Приложение позволяет углубить понимание структуры атомов, взаимодействия частиц и ядер, а также их энергетических и структурных свойств. Результаты расчетов резерфордовских сечений могут быть применены в различных научных и инженерных областях, включая ядерную энергетику, медицину, материаловедение и другие.

Дальнейшее развитие приложения может включать расширение его функциональности и возможностей. Например, можно добавить поддержку расчетов резерфордовских сечений для более сложных систем, включая молекулы и экзотические ядра. Также можно усовершенствовать алгоритмы и методы расчета, чтобы повысить эффективность и точность приложения.

В целом, разработка компьютерного приложения для расчета резерфордовских сечений открывает новые возможности для исследований и развития в области физики элементарных частиц и ядерной физики. Это инструмент, который позволяет более глубоко понять и описать взаимодействие частиц и ядер, способствуя развитию науки и применению полученных знаний в различных сферах человеческой деятельности.

Создана компьютерная программа для расчета кинетической энергии продуктов ядерных реакций на основе языков программирования Python и Java.

С помощью кинематического анализа с высокой точностью определен качественный состав формируемых спектров частиц. Для ускорения вычислений и для сравнения также использовался язык программирования Java. Данные, полученные по результатам исследований на ускорителе, были введены в программу и проведены расчеты. Результат открывался в отдельном окне, а диаграмма рассеяния отображалась в виде графика.

Графический интерфейс программы создан с помощью инструментария PyQt5 в кроссплатформенной среде QT-Designer.

Сравнивались времена вычислений оптимизированных кодов на языках Python и Java. Было обнаружено, что оба языка имеют малое время вычислений и не имеют существенной разницы в скорости.

Список литературы

2021. K.N. Mukhin. Experimental nuclear physics. Vol.1. М.: Изд-во «Лань» K.N. Mukhin. Experimental nuclear physics. Vol.2. М.: Изд-во «Лань» 2021. 6 с.
1. Блан Д. Ядра, частицы, ядерные реакторы: Пер. с франц. -М.: «Мир», 1989. -336 с., нл.
 2. А. М. Балдин, В. И. Гольданский, И. Л. Розенталь. Кинематика ядерных реакции. «Государственное издательство физико-математической литературы», 1959.
 3. Dubovichenko S., Burtebayev N., Dzhazairov-Kakhramanov A., Zazulin D., Kerimkulov Zh., Nassurlla M., Omarov Ch., Tkachenko A., Shmygaleva T., Kliczewski S. New measurements and phase shift analysis of $p^{16}\text{O}$ elastic scattering at astrophysical energies // Chinese Physics C. – 2017. – Vol. 41. – No. 1. – P. 014001(10).
 4. Burtebayev N., Sakhiyev S.K., Janseitov D.M., Kerimkulov Zh., Alimov D. Investigation of the elastic and inelastic scattering of α -particles from ^{13}C in the energy range 26.6–65 MeV // International Journal of Modern Physics E. – 2016. – Vol. 25. – No. 10. – P. 1650078(12).
 5. Burtebayev N., Janseitov D. M., Kerimkulov Zh., Mukhamejanov Y. S., Nassurlla M., Demyanova A. S., Danilov A. N., Ogloblin, A. A., Aimaganbetov A. S. Investigation of exotic states of ^{13}C at low energy // International Journal of Modern Physics E. – 2018. – Vol. 27. – No. 3. – P. 1850025(7)
 6. Burtebayev N., Burtebayeva J. T., Duisebayev A., Kerimkulov Zh. K., Nassurlla M., Zholdybayev T., Artemov S. V., Karakhodzhayev A. A., Salikhbayev U. S., Sakuta S. B., Kliczewski S., Piasecki E., Rusek K., Siudak R., Trzcinska A., Wolinska-Cichocka M., Amar A. Mechanizm of the $^7\text{Li}(d,t)^6\text{Li}$ reaction at 25 MeV energy of deuterons, values of spectroscopic factors and asymptotic normalization coefficients for the $\text{Li-7} \rightarrow \text{Li-6} + n$ vertex // Acta Physica Polonica B. – 2015. – Vol. 46. – No. 5. – P. 1037-1054.
 7. Nassurlla M., Burtebayev N., Janseitov D.M., Kerimkulov Zh., Alimov D., Morzabayev A.K., Talpakova K., Mukhamejanov Y., Galanina L.I., Demyanova A.S., Danilov A.N., Starastin V. Study of elastic and inelastic scattering of deuterons by ^9Be at energy $E = 14.5$ MeV // Acta Physica Polonica B. – 2020. – Vol.51, - No.3. – P. 751-756
 8. Распространенность изотопов в природе и их масса, точная таблица.
<https://dpva.ru/Guide/GuideChemistry/PeriodicalMendeleevTable/IsotopiAbundance> 1/
 9. Centre for photonuclear experiments data.

<http://cdfesinp.msu.ru/services/ensdfr.html>

10. Реакции. Характеристики вероятности реакций.

http://nuclphys.sinp.msu.ru/nucmodl/nml11_1.htm Ускорители заряженных частиц.

<https://www.booksite.ru/fulltext/1/001/008/114/616.htm>

11. Particle accelerator.

<https://www.accessscience.com/content/particle-accelerator/490900>

12. Plotting with PyQtGraph.

<https://www.mfitzp.com/tutorials/plotting-pyqtgraph/>

13. Kerimkulov Zh., Burtebayev N., Burtebayeva J.T., Duisebayev A., Nassurlla M., Zholdybayev T., Artemov S.V., Karakhodzhayev A.A., Salikhbayev U.S., Sakuta S.B., Kliczewski S., Piasecki E., Rusek K., Siudak R., Trzcińska A., Wolińska-Cichocka M., Amar A. Mechanism of the ${}^7\text{Li}(d,t){}^6\text{Li}$ reaction at 25 MeV energy of deuterons, values of spectroscopic factors and asymptotic normalization coefficients for the ${}^7\text{Li} \rightarrow {}^6\text{Li} + n$ vertex // Acta Physica Polonica B. – 2015. – Vol. 46. – No. 5. – P. 1037-1054

14. Dubovichenko S.B., Kerimkulov Zh.K., Burtebaev N., Zazulin D.M.,

Amar A. S. A. Astrophysical S-Factor for the Radiative-Capture Reaction $p{}^6\text{Li} \rightarrow {}^7\text{Be}\gamma$

// Physics of Atomic Nuclei. – 2011. – Vol. 74. – No. 7. – P. 984–1000

15. Физика атома и атомного ядра.

<https://physics.ru/courses/op25part2/content/chapter6/section/paragraph8/theory.html#.YKuaci1eO9a>

16. Г.И. Бабат. Ускорители. Молодая гвардия. 1957 г.

17. В. Е. Бунаков. Кинетические уравнения в теории ядерных реакций.

«Физика элементарных частиц и атомного ядра», 1980, ТОМ 11, Вып. 6

Приложение А

Код программы на языке Python

```
import sys
import
math
import
matplotlib
matplotlib.use('Qt5Agg')
from PyQt5 import QtCore, QtGui, QtWidgets
from matplotlib.backends.backend_qt5agg import FigureCanvasQTAgg,
NavigationToolbar2QT as NavigationToolbar
from matplotlib.figure import
Figurefrom pdata import
_____init___
class
  Ui_MainWindow(object):
  def setupUi(self,
  MainWindow):
    MainWindow.setObjectName("MainWindow")
    MainWindow.resize(511, 376)
    MainWindow.setStyleSheet("background-color: rgb(255, 255,
    255);")self.centralwidget = QtWidgets.QWidget(MainWindow)
    self.centralwidget.setStyleSheet("\n"
"background-color: rgb(228, 210, 166);")
    self.centralwidget.setObjectName("centralwidget"
) self.Basu =
    QtWidgets.QPushButton(self.centralwidget)
    self.Basu.setGeometry(QtCore.QRect(90, 340,
    113, 32))font = QtGui.QFont()
    font.setFamily(".PingFang
    SC") font.setPointSize(20)
    font.setBold(False)
    font.setItalic(False)
    font.setUnderline(False)
    font.setWeight(50)
    font.setStrikeOut(False)
    font.setKerning(True)
    self.Basu.setFont(font)
    self.Basu.setMouseTracking(Fa
    lse)
    self.Basu.setStyleSheet("background-color: rgb(3, 0, 1);\n"
"border-radius: 15px;\n"
"color: white;")
    self.Basu.setObjectName("Basu")
```

```

self.gridLayoutWidget =
QtWidgets.QWidget(self.centralwidget)
self.gridLayoutWidget.setGeometry(QtCore.QRect(10, 0, 321,
345))
self.gridLayoutWidget.setObjectName("gridLayoutWidget")
self.gridLayout =
QtWidgets.QGridLayout(self.gridLayoutWidget)
self.gridLayout.setContentsMargins(0, 0, 0, 0)
self.gridLayout.setObjectName("gridLayout")
self.label_8 = QtWidgets.QLabel(self.gridLayoutWidget)
font = QtGui.QFont()
font.setBold(True)
font.setWeight(75)
self.label_8.setFont(font)
self.label_8.setObjectName("label_8")
self.gridLayout.addWidget(self.label_8, 6, 0, 1,
1) self.e =
QtWidgets.QLineEdit(self.gridLayoutWidget)
font = QtGui.QFont()
font.setPointSize(15)
font.setBold(False)
font.setItalic(False)
font.setWeight(50)
self.e.setFont(font)
self.e.setObjectName("e")
self.gridLayout.addWidget(self.e, 8, 1, 1, 1)
self.label_5 =
QtWidgets.QLabel(self.gridLayoutWidget)font =
QtGui.QFont()
font.setBold(True)
font.setWeight(75)
self.label_5.setFont(font)
self.label_5.setObjectName("label_5")
self.gridLayout.addWidget(self.label_5, 3, 0, 1, 1)
self.label_6 =
QtWidgets.QLabel(self.gridLayoutWidget)font =
QtGui.QFont()
font.setBold(True)
font.setWeight(75)
self.label_6.setFont(font)
self.label_6.setObjectName("label_6")
self.gridLayout.addWidget(self.label_6, 4, 0, 1, 1)
self.label_9 =
QtWidgets.QLabel(self.gridLayoutWidget)font =

```

```

QtGui.QFont()
font.setBold(True)
font.setWeight(75)
self.label_9.setFont(font)
self.label_9.setObjectName("label_9")
self.gridLayout.addWidget(self.label_9, 7, 0, 1, 1)
self.label_4 =
QtWidgets.QLabel(self.gridLayoutWidget)font =
QtGui.QFont()
font.setBold(True)
font.setWeight(75)
self.label_4.setFont(font)
self.label_4.setObjectName("label_4")
self.gridLayout.addWidget(self.label_4, 2, 0, 1, 1)
self.DM2 =
QtWidgets.QLineEdit(self.gridLayoutWidget)font =
QtGui.QFont()
font.setPointSize(15)
self.DM2.setFont(font)
self.DM2.setObjectName("DM2")
self.gridLayout.addWidget(self.DM2, 5, 1, 1, 1)
self.M2 =
QtWidgets.QLineEdit(self.gridLayoutWidget)font
= QtGui.QFont()
font.setPointSize(15)
self.M2.setFont(font)
self.M2.setObjectName("M2")
self.gridLayout.addWidget(self.M2, 1, 1, 1, 1)
self.M3 =
QtWidgets.QLineEdit(self.gridLayoutWidget)font
= QtGui.QFont()
font.setPointSize(15)
self.M3.setFont(font)
self.M3.setObjectName("M3")
self.gridLayout.addWidget(self.M3, 2, 1, 1, 1)
self.label_3 =
QtWidgets.QLabel(self.gridLayoutWidget)font =
QtGui.QFont()
font.setBold(True)
font.setWeight(75)
self.label_3.setFont(font)
self.label_3.setObjectName("label_3")
self.gridLayout.addWidget(self.label_3, 1, 0, 1, 1)
self.label_2 =

```

```

QtWidgets.QLabel(self.gridLayoutWidget)font =
QtGui.QFont()
font.setPointSize(13)
font.setBold(True)
font.setWeight(75)
self.label_2.setFont(font)
self.label_2.setObjectName("label_2")
self.gridLayout.addWidget(self.label_2, 0, 0, 1, 1)
self.M4 =
QtWidgets.QLineEdit(self.gridLayoutWidget)font
= QtGui.QFont()
font.setPointSize(15)
self.M4.setFont(font)
self.M4.setObjectName("M4")
self.gridLayout.addWidget(self.M4, 3, 1, 1, 1)
self.M1 = QtWidgets.QLineEdit(self.gridLayoutWidget)
self.M1.setEnabled(True)

font = QtGui.QFont()
font.setPointSize(15)
self.M1.setFont(font)
self.M1.setObjectName("M1")
self.gridLayout.addWidget(self.M1, 0, 1, 1, 1)
self.label_10 =
QtWidgets.QLabel(self.gridLayoutWidget)font =
QtGui.QFont()
font.setBold(True)
font.setWeight(75)
self.label_10.setFont(font)
self.label_10.setObjectName("label_10")
self.gridLayout.addWidget(self.label_10, 8, 0, 1, 1)
self.label_7 =
QtWidgets.QLabel(self.gridLayoutWidget)font =
QtGui.QFont()
font.setBold(True)
font.setWeight(75)
self.label_7.setFont(font)
self.label_7.setObjectName("label_7")
self.gridLayout.addWidget(self.label_7, 5, 0, 1, 1)
self.DM4 =
QtWidgets.QLineEdit(self.gridLayoutWidget)font =
QtGui.QFont()
font.setPointSize(15)
self.DM4.setFont(font)
self.DM4.setObjectName("DM4")

```

```

self.gridLayout.addWidget(self.DM4, 7, 1, 1, 1)
self.DM3 =
QtWidgets.QLineEdit(self.gridLayoutWidget)font =
QtGui.QFont()
font.setPointSize(15)
self.DM3.setFont(font)
self.DM3.setObjectName("DM3")
self.gridLayout.addWidget(self.DM3, 6, 1, 1, 1)
self.DM1 = QtWidgets.QLineEdit(self.gridLayoutWidget)
font = QtGui.QFont()
font.setPointSize(15)
self.DM1.setFont(font)
self.DM1.setObjectName("DM1")
self.gridLayout.addWidget(self.DM1, 4, 1, 1, 1)
self.label_11 =
QtWidgets.QLabel(self.gridLayoutWidget)font =
QtGui.QFont()
font.setBold(True)
font.setWeight(75)
self.label_11.setFont(font)

self.label_11.setObjectName("label_11")
self.gridLayout.addWidget(self.label_11, 9, 0, 1, 1)
self.label_12 =
QtWidgets.QLabel(self.gridLayoutWidget)font =
QtGui.QFont()
font.setBold(True)
font.setWeight(75)
self.label_12.setFont(font)
self.label_12.setObjectName("label_12")
self.gridLayout.addWidget(self.label_12, 10, 0, 1, 1)
self.ktet =
QtWidgets.QLineEdit(self.gridLayoutWidget)font =
QtGui.QFont()
font.setPointSize(16)
self.ktet.setFont(font)
self.ktet.setText("")
self.ktet.setObjectName("ktet")
self.gridLayout.addWidget(self.ktet, 9, 1, 1, 1)
self.ku = QtWidgets.QLineEdit(self.gridLayoutWidget)
font = QtGui.QFont()
font.setPointSize(16)
self.ku.setFont(font)
self.ku.setText("")
self.ku.setObjectName("ku")

```

```

self.gridLayout.addWidget(self.ku, 10, 1, 1, 1)
self.result =
QtWidgets.QTextBrowser(self.centralwidget)
self.result.setGeometry(QtCore.QRect(330, 30, 181,
271))font = QtGui.QFont()
font.setFamily("Times New
Roman")font.setPointSize(14)
font.setBold(False)
font.setItalic(False)
font.setWeight(50)
self.result.setFont(font)
self.result.setStyleSheet("background-color: rgb(225, 201, 143);\n"
"font: 14pt \"Times New Roman\";")
self.result.setObjectName("result
")
self.gra =
QtWidgets.QPushButton(self.centralwidget)
self.gra.setGeometry(QtCore.QRect(360, 300,
113, 32))self.gra.setObjectName("gra")
self.Qwrite = QtWidgets.QLabel(self.centralwidget)
self.Qwrite.setGeometry(QtCore.QRect(390, 0, 121, 29))
self.Qwrite.setStyleSheet("background-color: rgb(216, 187,
115);")self.Qwrite.setText("")
self.Qwrite.setObjectName("Q
write") font = QtGui.QFont()
font.setPointSize(19)
font.setBold(True)
font.setWeight(75)
MainWindow.setCentralWidget(self.centralwidget)
self.retranslateUi(MainWindow)
QtCore.QMetaObject.connectSlotsByName(MainWindow)
def retranslateUi(self, MainWindow):
_translate = QtCore.QCoreApplication.translate
MainWindow.setWindowTitle(_translate("MainWindow", "KE
calculator"))self.Basu.setText(_translate("MainWindow", "Evaluate"))
self.label_8.setText(_translate("MainWindow", "Defect mass of
particle 3"))self.e.setText(_translate("MainWindow", "25.00000"))
self.label_5.setText(_translate("MainWindow", "Mass of particle 4"))
self.label_6.setText(_translate("MainWindow", "Defect mass of
particle 1"))self.label_9.setText(_translate("MainWindow", "Defect
mass of particle 4"))self.label_4.setText(_translate("MainWindow",
"Mass of particle 3")) self.DM2.setText(_translate("MainWindow",
"8.667700"))
self.M2.setText(_translate("MainWindow", "11.0093052"))

```



```

self.M3.setText(_translate("MainWindow", "4.0026033"))
self.label_3.setText(_translate("MainWindow", "Mass of
particle 2"))self.label_2.setText(_translate("MainWindow",
"Mass of particle 1"))
self.M4.setText(_translate("MainWindow", "9.0121829"))
self.M1.setText(_translate("MainWindow", "2.01410222"))
self.label_10.setText(_translate("MainWindow", "Initial kinetic
energy")) self.label_7.setText(_translate("MainWindow", "Defect
mass of particle 2"))self.DM4.setText(_translate("MainWindow",
"11.348100"))
self.DM3.setText(_translate("MainWindow", "2.424920"))
self.DM1.setText(_translate("MainWindow", "13.135900"))
self.label_11.setText(_translate("MainWindow", "Theta"))
self.label_12.setText(_translate("MainWindow", "Ku"))
self.result.setHtml(_translate("MainWindow", "<!DOCTYPE HTML
PUBLIC
\"-//W3C//DTD HTML 4.0//EN\"
\"http://www.w3.org/TR/REC-html40/strict.dtd\">\n"
"<html><head><meta name=\"qrichtext\" content=\"1\"
/><styletype=\"text/css\">\n"
"p, li { white-space: pre-wrap; }\n"
"</style></head><body style=\" font-family:\'Times New Roman\'; font-
size:14pt;font-weight:400; font-style:normal;\")>\n"
"<p style=\"-qt-paragraph-type:empty; margin-top:0px; margin-bottom:0px;
margin-
left:0px; margin-right:0px; -qt-block-indent:0; text-indent:0px;\")<br
/></p></body></html>"))
    self.gra.setText(_translate("MainWindow", "Show
graph"))
# Create application
app =
QtWidgets.QApplication(sys.argv)
MainWindow =
QtWidgets.QMainWindow()ui =
Ui_MainWindow()
ui.setupUi(MainWindow)
MainWindow.show()
li_1 = []
li_2 = []
def resultt():
    f1 =
    open("newname1.txt",
'w')f4 = open("new.txt",
'w')

```

```

m1 =
float(ui.M1.text())
m2 =
float(ui.M2.text())
m3 =
float(ui.M3.text())
m4 =
float(ui.M4.text())
dm1 =
float(ui.DM1.text())
dm2 =
float(ui.DM2.text())
dm3 =
float(ui.DM3.text())
dm4 =
float(ui.DM4.text())
e1 = float(ui.e.text())
kteta =
float(ui.ktet.text())k
= float(ui.ku.text())
q = (dm1 + dm2) - (dm3 + dm4) - k
v1 = m1 * m2 * e1
v2 = m3 + m4
v = v1 / (v2 * v2)
w = (((m4 - m1) * e1 + m4 * q) *
v2) / v1theta = 0
f3 = f1.write('Angle Energy \n')
while theta <= kteta:
    gr = (math.pi *
theta) / 180co =
math.cos(gr)
e3 = (co + math.sqrt(co * co + w)) ** 2 * v
if theta<10:
    f2 = (str(theta) + ' |' + str(round(e3, 2)) + ' MeV ' + ")
    f3 = f1.write(str(theta) + ' |' + str(round(e3, 2)) + ' MeV '
+ '\n')li_1.append(theta)
    li_2.append(round(e3, 3))
else:
    f2 = (str(theta) + ' |' + str(round(e3, 2)) + ' MeV ' + ")
    f3 = f1.write(str(theta) + ' |' + str(round(e3, 2)) + ' MeV '
+ '\n')li_1.append(theta)
    li_2.append(round(
e3, 3))theta = theta +
1

```

```

f4.write(','.join([str(i) for i in li_1]))
f4.write('\n')
f4.write(','.join([str(i) for i in
li_2]))f4.close()
def flo():
    dm1 =
float(ui.DM1.text())
    dm2 =
float(ui.DM2.text())
    dm3 =
float(ui.DM3.text())
    dm4 =
float(ui.DM4.text())
    k = float(ui.ku.text())
    f =
open("newname1.txt",
"r+")readdata = f.read()
    ui.result.setText(f'{readd
ata}') f.close()
    Q = (dm1 + dm2) - (dm3 + dm4) - k
    ui.Qwrite.setText("Q = " + str(round(Q, 4)))
ui.gra.clicked.connect(
_____i
nit
_____
)
ui.Basu.clicked.connect(r
esultt)
ui.Basu.clicked.connect(f
lo) sys.exit(app.exec_())

```

Бағдарламаның Java тіліндегі коды

```

package
com.company;
import java.io.*;
import
java.math.BigDecimal;
import
java.math.MathContext;
import
java.math.RoundingMode;
import java.nio.file.Files;
import
java.nio.file.Path;

```

```

import
java.nio.file.Paths;
import
java.util.ArrayList;
import
java.util.List;
import
java.util.Scanner;
public class Main {
    public static void case1() {
        Scanner scan = new Scanner(System.in);
    }
    public static void main(String[]
    args) {File myFile = new
    File("date.txt"); Scanner
    myReader = null;
    Scanner scanner = new Scanner(System.in);
    List<BigDecimal> fList = new
    ArrayList<BigDecimal>();try {
        myReader = new Scanner(myFile);
    } catch (FileNotFoundException e) {
        e.printStackTrace();
    }
    while
    (myReader.hasNextLine()) {
        String data =
        myReader.nextLine();
        fList.add(new
        BigDecimal(data));
    }
    myReader.close();
    System.out.println(fList);

    BigDecimal m1 =
    fList.get(0); BigDecimal
    m2 = fList.get(1);
    BigDecimal m3 =
    fList.get(2); BigDecimal
    m4 = fList.get(3);
    BigDecimal dm1 =
    fList.get(4);BigDecimal
    dm2 = fList.get(5);
    BigDecimal dm3 =
    fList.get(6);BigDecimal

```

```
dm4 = fList.get(7);
BigDecimal e1 =
fList.get(8);
```

```
System.out.print("Есептеу бұрышы
= "); int kteta = scanner.nextInt();
System.out.print("Қозу деңгейі = ");
BigDecimal ku =
scanner.nextBigDecimal();
System.out.print("Есептейміз бе?");
char c = scanner.next().charAt(0);
```

```
if (c == 'y') {
    System.out.println("Жаңа файлға шығарамыз ба?");
    char D =
scanner.next().charAt(0);
    while (D == 'y') {
        System.out.print("Жаңа файл
атауы:");String newName =
scanner.next();
        Path path = Paths.get("data" + newName + ".txt"); //creates Path
instance
        try {
            Path p = Files.createFile(path);
            BufferedWriter writer = new
            BufferedWriter(new FileWriter(p.toString()));

            BigDecimal q =
            ((dm1.add(dm2)).subtract(dm3.add(dm4))).subtract(ku);
            MathContext m = new
            MathContext(6); writer.write("Q = "
            + q.round(m)); writer.write("Қозу
деңгейі = " + ku + "\n");
            writer.write("Бұрыш |");
            writer.write("Энергия" + "\n");
            writer.write("_____ " +
            "\n");

            BigDecimal v1 =
            m1.multiply(m3).multiply(e1);BigDecimal
            v2 = m3.subtract(m4);
            BigDecimal v = v1.divide(v2.multiply(v2),2
            ,
RoundingMode.HALF_UP);
            BigDecimal w
```



```
NavigationToolbar2QT as
NavigationToolbarfrom
matplotlib.figure import Figure
```

```
with open("new.txt", "r") as f:
    readdata = f.read()
    ff = [str(x) for x in
readdata.split()]m=[int(s) for s
in ff[0].split(',')]
n=[float(s) for s in
ff[1].split(',')]print(m)
print(n)
```

```
class MplCanvas(FigureCanvasQTAgg):
    def __init__(self, parent=None, width=5, height=4,
        dpi=100):fig = Figure(figsize=(width, height),
        dpi=dpi)
        self.axes = fig.add_subplot(111)
        super(MplCanvas, self).__init__(fig)
```

```
class MainWindow(QtWidgets.QMainWindow):
```

```
    def __init__(self, *args, **kwargs):
        super(MainWindow, self).__init__(*args, **kwargs)
```

```
        sc = MplCanvas(self, width=5, height=4,
            dpi=100)sc.axes.plot(m, n)
```

```
        # Create toolbar, passing canvas as first parament, parent (self,
theMainWindow) as second.
```

```
        toolbar = NavigationToolbar(sc,
self)layout =
        QtWidgets.QVBoxLayout()
        layout.addWidget(toolbar)
        layout.addWidget(sc)
```

```
        # Create a placeholder widget to hold our toolbar and canvas.
```

```
        widget = QtWidgets.QWidget()
        widget.setLayout(layout)
        self.setCentralWidget(widget)
        self.show()
```

```
app =
QtWidgets.QApplication(sys.argv)
w = MainWindow()
```

```
app.exec_()
```

pyqtgraph кітапханасының көмегімен график тұрғызу

```
кодыfrom PyQt5 import QtWidgets, QtCore
```

```
from pyqtgraph import
```

```
PlotWidget, plotimport pyqtgraph
```

```
as pg
```

```
import sys # We need sys so that we can pass argv to QApplication
```

```
import os
```

```
with open("new.txt",
```

```
    "r") as f:readdata =
```

```
    f.read()
```

```
    ff = [str(x) for x in readdata.split()]
```

```
m=[int(s) for s in ff[0].split(',')]
n=[float(s) for s in
```

```
ff[1].split(',')]print(m)
print(n)
```

```
class MainWindow1(QtWidgets.QMainWindow):
```

```
    def __init__(self, *args, **kwargs):
```

```
        super(MainWindow1, self).__init__(*args,
```

```
        **kwargs)self.graphWidget = pg.PlotWidget()
```

```
        self.setCentralWidget(self.graphWidget)
```

```
        self.graphWidget.setBackground('w')
```

```
        self.graphWidget.setTitle("<span style=\"color:blue;font-
size:30pt\">Angle/Energy dependence</span>")
```

```
        pen = pg.mkPen(color=(60, 0, 106), width=5, style=QtCore.Qt.SolidLine)
```

```
        self.graphWidget.plot(m, n, pen=pen, symbol='+', symbolSize=6,
```

```
symbolBrush=('r'))
```

```
        self.graphWidget.plot(m, n, pen=pen)
```

```
        self.graphWidget.showGrid(x=True,
```

```
        y=True) styles = {'color': 'b', 'font-size':
```

```
'30px'} self.graphWidget.setLabel('left',
```

```
'Angle', **styles)
```

```
        self.graphWidget.setLabel('bottom', 'Energy, MeV', **styles)
```

```
def main():
```

```
    app = QtWidgets.QApplication(sys.argv)
```

```
    main =
```

```
    MainWindow1()
```

```
    main.show()
```

```
    sys.exit(app.exec_
```

```
    ())
```

```
if __name__ == '__main__':
```

```
    main()
```