Министерство науки и высшего образования Российской Федерации

Федеральное государственное автономное образовательное учреждение

высшего образования

«Уральский федеральный университет

имени первого Президента России Б.Н. Ельцина»

Физико-Технологический Институт

Кафедра технической физики

Расчетно-графическая работа

«**РАЗРАБОТКА ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ ДЛЯ АВТОМАТИЗИРОВАННОГО АНАЛИЗА И ВИЗУАЛИЗАЦИИ ЛОГОВ МОДЕЛИРОВАНИЯ ЭЛЕМЕНТАРНЫХ ЧАСТИЦ В ПРОГРАММНОЙ СРЕДЕ GEANT4**»

|  |  |
| --- | --- |
| Преподаватель: | Новоселов Иван Эдуардович |
| Студенты: | Маркова Алёна Денисовна |
|  | Черняков Матвей Сергеевич |
|  | Артюшин Артём Александрович |
| Группа: | Фт-420008 |

Екатеринбург

2025

Содержание

[НОРМАТИВНЫЕ ССЫЛКИ 3](#_Toc207463934)

[УСЛОВНЫЕ ОБОЗНАЧЕНИЯ И СОКРАЩЕНИЯ 4](#_Toc207463935)

[ВВЕДЕНИЕ 5](#_Toc207463936)

[1 ПРОБЛЕМАТИКА ТОГО-ТО ТОГО-ТО, ЧЕГО-ТО ЧЕГО-ТО, ТАМ-ТО ТАМ-ТО 6](#_Toc207463937)

[1.1 Название подраздела 6](#_Toc207463938)

[1.2 Название следующего подраздела 8](#_Toc207463939)

[1.3 Следующий раздел 11](#_Toc207463940)

[2 МОДЕЛИРОВАНИЕ ТОГО-ТО ЧЕГО-ТО 13](#_Toc207463941)

[2.1 Такое-то моделирование 13](#_Toc207463942)

[2.1.1 Название пункта 13](#_Toc207463943)

[2.1.2 Название следующего пункта 14](#_Toc207463944)

[3 ПРОЕКТИРОВАНИЕ 15](#_Toc207463945)

[3.1 Внутреннее проектирование 15](#_Toc207463946)

[3.2 Внешнее проектирование 15](#_Toc207463947)

[4 ИНЖЕНЕРНАЯ РЕАЛИЗАЦИЯ 16](#_Toc207463948)

[ЗАКЛЮЧЕНИЕ 17](#_Toc207463949)

[СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ 18](#_Toc207463950)

[ПРИЛОЖЕНИЕ А. Схема электрическая принципиальная 2](#_Toc207463951)0

[ПРИЛОЖЕНИЕ Б. Технические характеристики элементов 21](#_Toc207463951)

[ПРИЛОЖЕНИЕ В. Техническое задание 22](#_Toc207463952)

[приложение Г. Акт испытания 28](#_Toc207463953)

[приложение Д. Акт о внедрении 29](#_Toc207463954)

[приложение Е. Сканы дипломов / наград и т.д. 30](#_Toc207463955)

НОРМАТИВНЫЕ ССЫЛКИ

В пояснительной записке использованы ссылки на следующие стандарты:

|  |  |
| --- | --- |
| **ГОСТ 19.701–90 ЕСПД** | Схема алгоритмов, программ, данных и систем. |
| **ГОСТ 34.602–89 ИТ. ЕКСАС** | Виды, комплектность и обозначение документов при создании автоматизированных систем. |
| **ГОСТ 34.602-2020** | Комплекс стандартов на автоматизированные системы.  Техническое задание на создание автоматизированной системы. |
| **ГОСТ 34.603–92 ит. ЕКСАС** | Виды испытаний автоматизированных систем. |
| **ГОСТ Р ИСО/МЭК 9126–93 ИТ** | Оценка программной продукции. Характеристика качества и руководства по их применению. |
| **ГОСТ Р ИСО/МЭК ТО 15910–2002 ИТ** | Процесс создания документации пользователя программного средства. |
| **ГОСТ Р ИСО/МЭК 12207–99 ИТ** | Процессы жизненного цикла программных средств. |
| **Р 50.1.028–2001 ИТПЖЦП** | Рекомендация по стандартизации. Методология функционального моделирования. |

условные обозначения и сокращения

|  |  |  |
| --- | --- | --- |
| **Geant4** | – | Библиотека для моделирования прохождения частиц через вещество |
| **CLI** | – | Command Line Interface – интерфейс командной строки |
| **GUI** | – | Graphical User Interface – графический пользовательский интерфейс |
| **CSV** | – | Comma-Separated Values – формат хранения табличных данных |
| **XLSX** | – | Формат электронных таблиц Microsoft Excel |
| **DF** | – | DataFrame – структура данных библиотеки pandas |
| **dE** | – | Energy Loss – потеря энергии частицы на шаге |
| **MeV** | – | Mega-electronvolt – мегаэлектронвольт |
| **mm** | – | Millimeter – миллиметр |
| Шаг (Step) | – | Минимальное продвижение частицы через вещество с учетом различных физических процессов |
| Трек (Track) | – | Полное продвижение частицы в веществе к моменту обращения к данному объекту |
| **Energy Deposit** | – | Энергия, осаждённая в материале (в MeV) |
| **Energy Leakage** | – | Энергия, покинувшая модель (утечка энергии) |

введение

Моделирование прохождения частиц через вещество с помощью программного пакета Geant4 является стандартом в ядерной физике, физике высоких энергий, радиобиологии и дозиметрии. ПО используется во многих исследовательских проектах по всему миру, в том числе и в России. Разработчиком является ЦЕРН (от фр. Conseil Europeen pour la Recherche Nucleaire, CERN) – Европейская организация по ядерным исследованиям. Выходными данными таких симуляций являются текстовые логи, содержащие детальную информацию о траекториях и взаимодействиях миллионов частиц. Однако с увеличением сложности моделей и использованием многопоточности для ускорения расчетов, ручной анализ этих логов становится практически невозможным.

Существующие инструменты для анализа выходных данных Geant4 часто требуют глубоких знаний специфических фреймворков и не предоставляют готовых средств для автоматического парсинга текстовых логов, их верификации и комплексной визуализации. Особенно остро стоит проблема сверки данных: итоговая сводка симуляции (Energy deposit, Process calls frequency) должна математически соответствовать сумме данных, извлеченных из пошаговой информации. Расхождения между ними могут указывать как на ошибки в парсинге, так и на тонкости работы самого движка Geant4, что требует тщательного расследования. Таким образом, существует потребность в создании специализированного программного обеспечения, способного автоматизировать извлечение, агрегацию, верификацию и визуализацию данных из многопоточных логов Geant4.

Основанием для разработки послужила конкретная задача анализа симуляций, реализованных в среде Geant4. Исходными данными являются текстовые логи, генерируемые симуляцией, которые включают:

1. многопоточные префиксы строк (например, G4WT17),
2. блоки пошаговой информации о треках частиц (координаты, энергия, потери на шаг, процессы и т.д.),
3. итоговые сводки симуляции (Energy deposit, Energy leakage, Process calls frequency).

**Объект исследования –** процесс анализа данных, генерируемых в результате моделирования в программном пакете Geant4.

Предмет исследования – методы и алгоритмы автоматического парсинга, верификации и визуализации многопоточных текстовых логов Geant4.

Цель исследования – разработка программного комплекса на языке Python для автоматизированного парсинга, анализа, верификации и визуализации данных из текстовых логов (включая многопоточные) симуляций, выполненных в среде Geant4, с последующей подготовкой исполняемого файла для операционной системы Windows и комплекта документации.

Для решения этой цели поставлены следующие задачи:

1. провести анализ структуры текстовых логов Geant4, включая форматы записи многопоточных префиксов, блоков шагов частиц и итоговых сводок, разработать методику верификации для сопоставления данных, извлеченных на уровне отдельных шагов, с агрегированными результатами в финальной сводке симуляции;
2. на основе проведенного анализа необходимо реализовать устойчивый алгоритм парсинга, способный распознавать и извлекать для каждого шага частицы такие данные, как идентификатор потока, координаты, кинетическую энергию, потерю энергии на шаг, длину шага и трека, объем, физический процесс, TrackID, ParentID и тип частицы, обеспечив при этом корректную обработку различных систем единиц измерения с их автоматическим приведением к стандартным величинам;
3. разработать модуль агрегации данных, обеспечивающий суммирование и статистический анализ по типам частиц, индивидуальным трекам и потокам выполнения, а также реализовать модуль верификации, выполняющий сравнение суммарных потерь энергии с величиной Energy Deposit из сводки и анализ расхождений, включая сверку частот процессов;
4. создать модуль визуализации, генерирующий набор графиков, включая гистограммы распределения кинетической энергии по типам частиц, графики распределения потерь энергии на шаг, диаграммы частот процессов и тепловые карты пространственного распределения координат;
5. реализовать графический интерфейс пользователя с возможностями загрузки файлов логов, настройки фильтров, запуска анализа и сохранения результатов, а также интерфейс командной строки для пакетной обработки, обеспечив при этом экспорт табличных данных в форматы CSV, XLSX и графиков в форматы SVG и PNG;
6. протестировать программу на различных входных логах, подготовку скрипта и инструкции по сборке проекта в исполняемый файл для Windows с использованием PyInstaller, а также оформление комплекта документации, включающего отчет с описанием методики и выводов, текст программы и презентацию для демонстрации функционала, в соответствии с требованиями ГОСТ и стандартов оформления ВКР.

1 Анализ структуры логов и проектирование системы

В данном разделе представлен аналитический обзор структуры логов симулятора Geant4, а также описаны методы проектирования и реализации системы автоматизированного парсинга и анализа данных. Целью является разработка архитектуры, обеспечивающей воспроизводимость, масштабируемость и достоверность результатов при работе с многопоточными моделями взаимодействия частиц. Рассмотрены особенности форматов логов, методики верификации корректности парсинга и общие принципы построения программных классов и алгоритмов обработки данных.

1.1 Форматы данных Geant4 и многопоточные префиксы

Система Geant4 формирует высокоструктурированные журналы вычислительных событий, включающие данные о материалах, геометрии модели, физических процессах и последовательностях взаимодействий частиц. Формат логов представляет собой последовательность текстовых блоков, разделённых по типам физических сущностей и этапам расчёта. Наиболее информативными фрагментами являются секции, содержащие описание треков частиц, энергетических потерь, путей в материале, а также итоговые сводки по энергии и частоте вызовов процессов.

В многопоточной конфигурации Geant4 каждому вычислительному потоку присваивается уникальный идентификатор, обозначаемый префиксом вида «G4WTXX». Этот префикс используется для маркировки каждой строки лога, что позволяет выделять и анализировать данные, относящиеся к конкретному потоку исполнения. Таким образом, обеспечивается возможность независимого сбора и последующей агрегации информации по потокам, что особенно важно при моделировании с использованием высокопроизводительных вычислительных узлов.

Каждый шаг частицы описывается набором параметров: координаты (x, y, z), кинетическая энергия (KineE), потеря энергии на шаге (dEStep), длина шага (StepLen), длина трека (TrakLen), наименование объёма и процесс взаимодействия. Идентификаторы Track ID и Parent ID позволяют дифференцировать первичные и вторичные частицы, что критически важно для последующего энергетического анализа.

В логах также присутствуют блоки метаданных, содержащие сведения о параметрах симуляции – используемых моделях, границах энергии, шагах и настройках взаимодействий (например, eIoni, eBrem, hIoni, Rayl и т.д.). Такая структура обеспечивает не только информативность, но и возможность программной верификации корректности работы физических моделей.

1.2 Методика верификации данных парсинга с итоговой сводкой

Разработанная методика верификации основана на сопоставлении параметров, извлечённых из логов, с контрольными значениями, указанными в итоговых разделах симуляции. Проверка корректности парсинга реализована на уровне как структуры данных, так и численных характеристик.   
Ключевой компонент системы – класс Geant4LogParser, который выполняет построчное чтение лога и идентификацию контекстных фрагментов по регулярным выражениям. Каждая строка проходит процедуру распознавания физических величин и их единиц измерения с последующим приведением к системе СИ (MeV, mm и т. д.). Данные каждого шага сохраняются в объект StepData, обеспечивающий строгую типизацию и проверку целостности. Для подтверждения достоверности парсинга применяется метод verify\_results(), сравнивающий агрегированные результаты по энергетическим потерям и количеству вызовов физических процессов с контрольными данными из сводных таблиц лога (в частности, разделы «Energy deposit» и «Process calls frequency»). Рассчитываются абсолютные и относительные отклонения, что позволяет оценить степень согласованности парсера с физическим движком Geant4.

Дополнительно формируются отчёты, включающие детализированную статистику по первичным (Parent ID = 0) и вторичным взаимодействиям, распределения кинетических энергий, а также энерго-балансовые соотношения. Для визуальной проверки предусмотрены построения графиков распределений и энергетических балансов, что облегчает контроль качества анализа.

1.3 Проектирование классов и алгоритмов обработки

Проектирование системы обработки логов выполнено с опорой на принципы модульности, расширяемости и повторного использования компонентов. Архитектура реализована в виде набора взаимосвязанных классов, обеспечивающих полный цикл анализа: от чтения данных до формирования визуальных и табличных отчётов.

Класс StepData представляет собой dataclass-структуру, описывающую отдельный шаг частицы. Он хранит координаты, энергетические параметры, длины пути, сведения о процессе и идентификаторы треков. Наличие свойства is\_primary() позволяет быстро классифицировать события по принадлежности к первичным трекам.

Класс Geant4LogParser реализует алгоритмы регулярного парсинга и нормализации данных. В процессе работы он строит таблицу шагов частиц, конвертирует единицы измерения и собирает сводные параметры, такие как суммарная энергия потерь и частотные характеристики процессов. Алгоритмы обработки включают фильтрацию по контекстным шаблонам, преобразование строковых величин в числовые значения и верификацию структурной целостности каждого шага.

Класс Geant4Analyzer выполняет статистический анализ собранных данных, включая группировку по типам частиц, процессам и потокам, а также вычисление агрегированных показателей (средние значения, стандартные отклонения, энергетические суммарные балансы). Реализованы функции визуализации, использующие библиотеки matplotlib и seaborn для построения диаграмм распределений, тепловых карт и корреляционных графиков.

Результаты анализа экспортируются в форматы CSV и XLSX, что обеспечивает совместимость с внешними инструментами обработки (Excel, R, MATLAB). Кроме того, предусмотрено создание отчётов в текстовом виде, содержащих ключевые метрики и результаты верификации энергетического баланса.

Таким образом, архитектура системы обеспечивает воспроизводимость анализа, автоматизацию проверки корректности данных и возможность интеграции с высокопроизводительными вычислительными конвейерами.

2 Реализация парсера и модуля агрегации данных

Во втором разделе рассмотрены вопросы практической реализации системы анализа логов Geant4. В рамках проекта была разработана модульная архитектура, включающая парсер логов, конвертер физических единиц, систему агрегации данных и подсистему верификации результатов. Основное внимание уделено методам обеспечения корректности обработки многопоточных логов, точности сопоставления энергетических параметров и воспроизводимости результатов на уровне агрегированных метрик. Раздел описывает программные алгоритмы, применяемые при разборе и нормализации данных, а также подходы к последующему статистическому анализу и контролю достоверности.

2.1 Разработка алгоритма парсинга, конвертера единиц и обработки многопоточности

Алгоритм парсинга построен на принципе последовательного анализа строк лога с использованием регулярных выражений для идентификации ключевых элементов: идентификаторов потоков, треков, родительских частиц, а также физических величин, характеризующих шаг взаимодействия. Каждая строка, содержащая данные о шаге частицы, подвергается разбору с выделением параметров: номера шага, координат (x, y, z), кинетической энергии, потери энергии dE, длины шага и типа физического процесса.

Особое внимание уделено корректной интерпретации многопоточной структуры логов Geant4. Поскольку при параллельных вычислениях каждая нить исполнения помечается префиксом формата «G4WTXX», в алгоритме реализовано динамическое определение текущего контекста потока. Это позволяет сохранять независимость данных, полученных из разных потоков, и исключать пересечение идентификаторов треков при агрегации.

Модуль конвертации единиц реализует автоматическое приведение физических параметров к единой системе – в частности, энергия пересчитывается в МэВ, длины – в миллиметры. Таблицы соответствий (eV – MeV, nm – mm и др.) закреплены в виде словарей, используемых на этапе обработки каждой строки. Такой подход исключает неоднозначности при анализе смешанных форматов и обеспечивает единообразие последующих вычислений.

Реализация парсера выполнена на языке Python с использованием стандартных библиотек re, pathlib и dataclasses, что позволило обеспечить читаемость и гибкость кода, а также возможность масштабирования под новые типы логов.

2.2 Агрегация данных по типам частиц и трекам

После этапа парсинга данные преобразуются в структурированный вид и помещаются в DataFrame библиотеки pandas, что обеспечивает эффективное выполнение агрегирующих и статистических операций. Для каждой частицы собираются сводные метрики: количество шагов, средняя и суммарная потеря энергии, диапазон кинематических характеристик и частота встречаемости физических процессов.

Агрегация выполняется в двух разрезах:

1. по типу частицы (particle): для анализа энергетических распределений и вклада различных частиц в общие потери энергии;
2. по трекам (track\_id): для выявления индивидуальных траекторий и локализации потерь энергии вдоль пути движения.

Реализация агрегатора предусматривает фильтрацию данных по признаку первичности (Parent ID = 0), что позволяет формировать отдельные сводки для первичных и вторичных взаимодействий. Дополнительно строятся распределения по процессам, позволяющие оценить интенсивность протекания ключевых механизмов взаимодействия (например, ионизации, тормозного излучения, рассеяния и т.д.).

Результирующие таблицы агрегатов используются для построения диаграмм плотности, гистограмм и сравнительных графиков. Это даёт возможность анализировать поведение частиц в материале, оценивать долю вклада различных процессов и проводить межсерийное сравнение симуляций.

2.3 Верификация Energy Deposit и частот процессов

На заключительном этапе анализа выполняется верификация энергетического баланса и частот вызова физических процессов. Данный этап служит инструментом контроля корректности как работы парсера, так и исходных симуляционных данных Geant4.

Методика основана на сравнении рассчитанных значений суммарных потерь энергии с данными из раздела «Energy deposit» в исходных логах. Абсолютное расхождение рассчитывается по формуле:

,

где – сумма положительных значений dEStep (потерь энергии), полученная в результате парсинга всех шагов частиц;

– значение Energy Deposit из итоговой сводки лога Geant4 (эталонное значение, рассчитанное самим Geant4).

Относительные расхождения рассчитывалось по формуле:

,

что позволяет оценивать погрешность в процентах от эталонного значения.

Дополнительно проводится сравнение частоты вызовов процессов, фиксируемых в парсере, с итоговой сводкой раздела «Process calls frequency». Для каждого процесса (например, eIoni, hIoni, Rayl и др.) вычисляются разности между подсчитанным и эталонным числом вызовов, а также формируется таблица отклонений. Результаты верификации сохраняются в структуре отчёта и могут быть визуализированы в виде бар-графов расхождений.

Проверка энергетического баланса также включает анализ распределения энергии между первичными и вторичными треками. На основе этих данных оценивается степень замыкания энергетического контура и достоверность моделирования. В случае превышения порогов допустимых расхождений предусмотрено автоматическое уведомление о несоответствии, что делает систему пригодной для автономного контроля качества симуляций.

3 Визуализация и пользовательские интерфейсы

В рамках разработки программного комплекса для анализа данных моделирования в среде Geant4 были реализованы два типа пользовательских интерфейсов: интерфейс командной строки (CLI) и графический интерфейс (GUI) на основе библиотеки Tkinter. Оба интерфейса обеспечивают гибкость взаимодействия с системой в зависимости от потребностей пользователя — от пакетной обработки данных до интерактивного анализа. Кроме того, система включает модуль визуализации, который предоставляет набор графиков и диаграмм для анализа распределения энергии, потерь на шаге, частоты процессов и пространственного распределения частиц.

3.1 Интерфейс командной строки

Интерфейс командной строки реализован с использованием модуля argparse стандартной библиотеки Python. Он предназначен для автоматизированного выполнения задач парсинга, анализа и экспорта данных без необходимости взаимодействия с графическим интерфейсом. Это особенно полезно при обработке больших объемов данных в пакетном режиме, а также при интеграции системы в скрипты обработки или CI/CD-конвейеры.

Основные функциональные возможности CLI:

* Задание входного файла лога через аргумент -i или –input;
* Выбор директории для выходных данных с помощью аргумента -o или –output;
* Настройка форматов экспорта данных (CSV, XLSX) и графиков (PNG, SVG) через аргументы --export и –plot;
* Режим отладки (--debug) для детализированного вывода процесса парсинга;
* Отключение визуализации (--no-viz) для ускорения обработки.

CLI обеспечивает полный цикл обработки: от чтения и парсинга лог-файла до генерации отчётов и визуализаций. Все этапы выполняются последовательно, с выводом текущего статуса в консоль, что позволяет отслеживать прогресс и выявлять возможные ошибки на ранних стадиях.

3.2. Графический интерфейс на Tkinter

Для пользователей, предпочитающих интерактивную работу с данными, был разработан графический интерфейс на основе библиотеки Tkinter. GUI предоставляет удобный доступ ко всем функциям системы через интуитивно понятные элементы управления: кнопки, выпадающие списки, поля ввода и область предпросмотра результатов.

Структура графического интерфейса включает:

* Область выбора файла – кнопка для загрузки лог-файла через системный диалог;
* Настройки экспорта – чекбоксы для выбора форматов данных и графиков;
* Прогресс-бар – визуализация хода выполнения анализа;
* Текстовая область – вывод логов выполнения и финального отчёта;
* Область предпросмотра графиков – отображение сгенерированных диаграмм непосредственно в интерфейсе.

Интерфейс реализован с использованием многопоточности: обработка данных выполняется в фоновом потоке, что предотвращает «зависание» интерфейса при работе с большими файлами. По завершении анализа пользователь получает возможность просмотреть графики, изучить отчёт и сохранить результаты в выбранной директории.

3.3. Гистограммы энергии, графики потерь dE, диаграммы процессов и тепловые карты

Визуализация является ключевым компонентом системы, позволяя пользователю интерпретировать результаты моделирования наглядно. Для этого реализован набор графиков, охватывающих основные аспекты симуляции:

Для каждого типа частиц (первичные и вторичные) строятся гистограммы распределения кинетической энергии (см. рисунок \*). На графиках отображаются:

* Количество шагов в каждом энергетическом интервале;
* Среднее значение энергии (вертикальная пунктирная линия);
* Статистические показатели: минимум, максимум, стандартное отклонение.

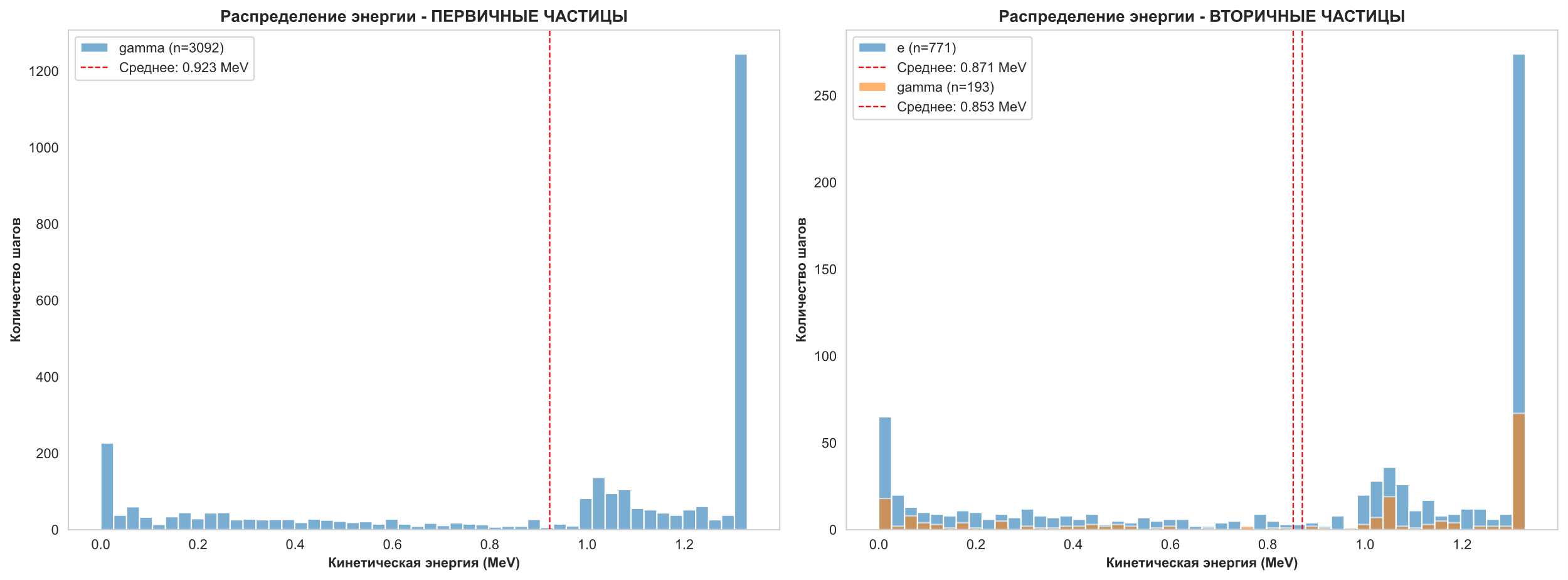


Рисунок \* – Гистограммы распределения кинетической энергии

Такие гистограммы позволяют оценить энергетический спектр частиц и выявить аномалии в распределении.

Для анализа энергетических потерь используется комбинация violin-plot и box-plot, которые отображают распределение значений dE для каждого типа частиц (см. рисунок \*). Особенности визуализации:

* Исключаются нулевые и отрицательные значения dE;
* Отображаются медиана, квартили и «усы» — диапазоны типичных значений;
* Цветовое кодирование по типам частиц.

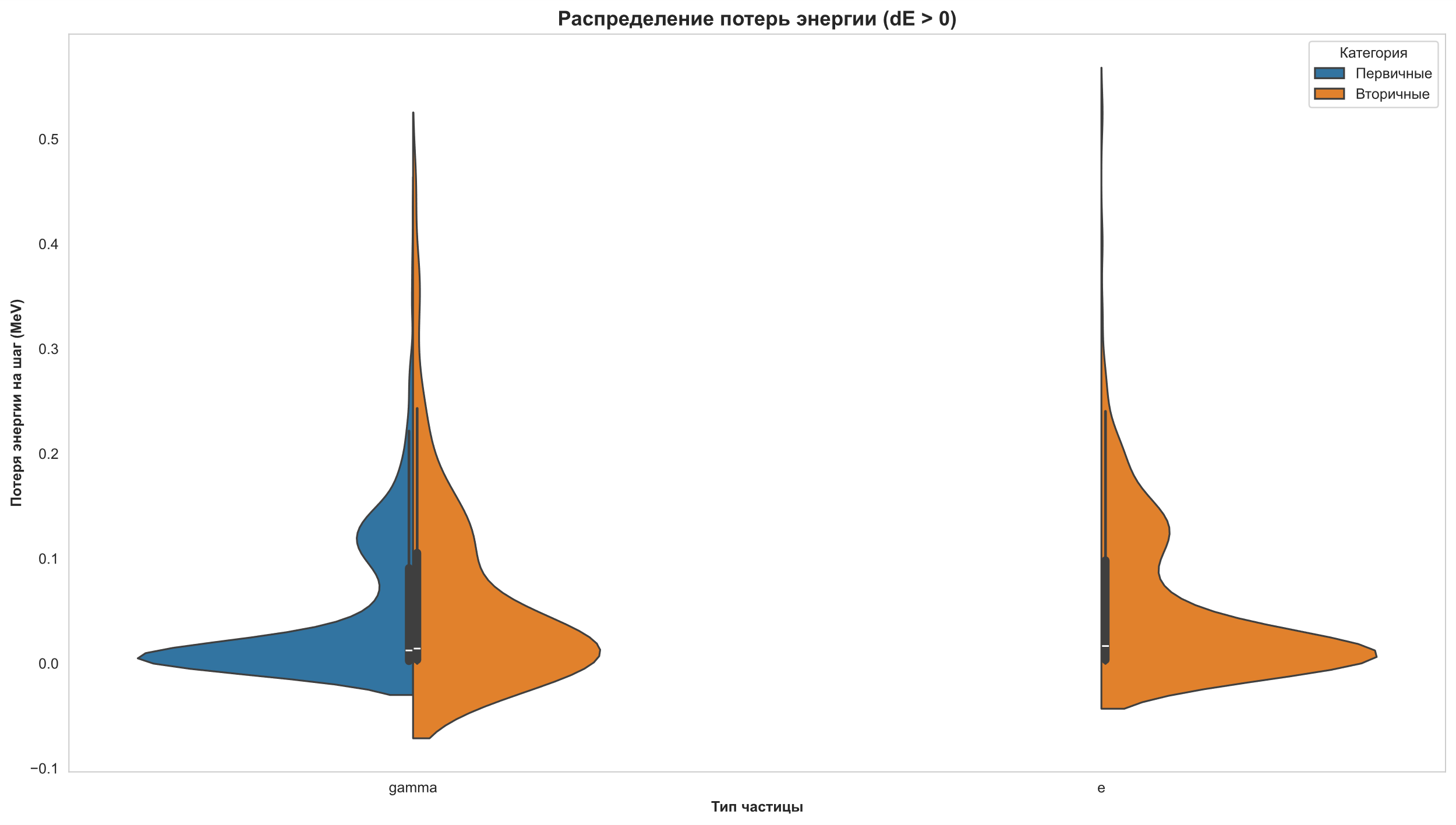


Рисунок \* – Графики энергетических потерь

Это позволяет оценить вклад различных частиц в общие энергетические потери.

Также строятся горизонтальные столбчатые диаграммы, отображающие частоту вызовов физических процессов (например, ионизация, комптоновское рассеяние, тормозное излучение). Для каждого процесса указывается абсолютное количество вызовов, что помогает идентифицировать доминирующие механизмы взаимодействия (см. рисунок \*).

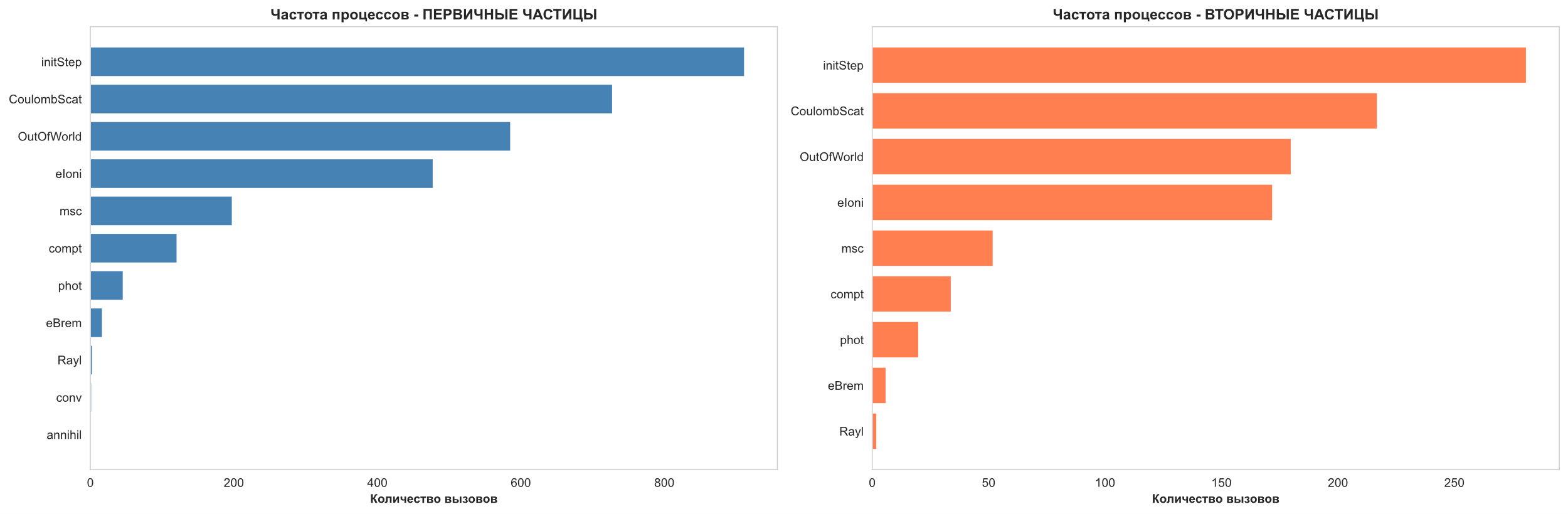


Рисунок \* – Диаграммы частоты вызовов процессов

Для визуализации плотности распределения частиц в пространстве используются двумерные гистограммы (heatmaps). Карты строятся для трёх проекций:

* XY;
* XZ;
* YZ.

Интенсивность цвета соответствует количеству частиц в данной области, что позволяет выявить зоны максимальной активности и пространственную структуру треков (см. рисунок \*).

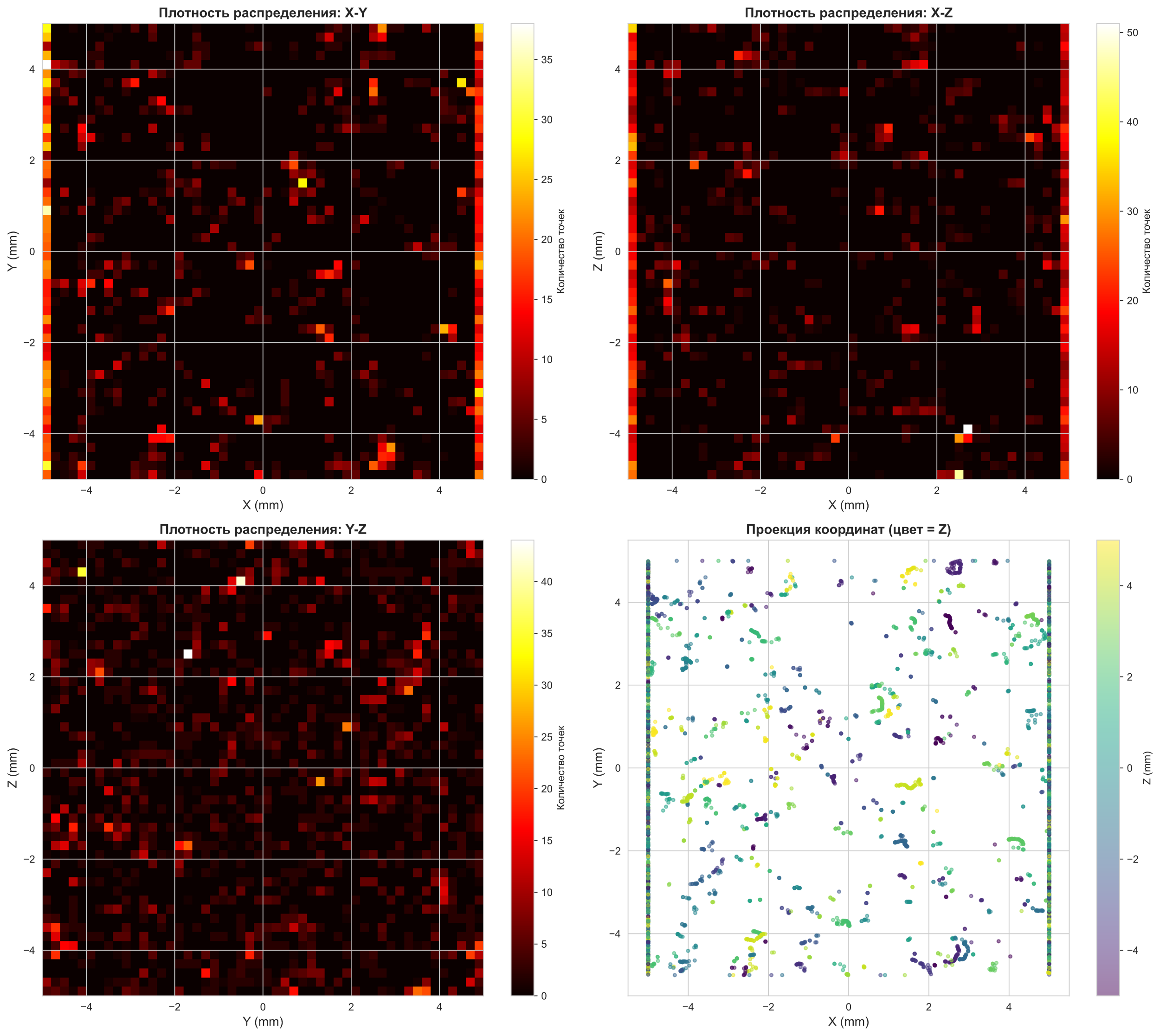


Рисунок \* – Графики плотности распределения частиц в пространстве

Дополнительно строятся сравнительные диаграммы, на которых отображаются:

* Сопоставление суммы dE из парсинга и значения Energy deposit из итоговой сводки;
* Сравнение частот процессов, полученных парсингом и из сводки;
* Распределение энергии для каждой отдельной частицы;
* Другая системная информация из логов.

Эти графики служат для проверки корректности парсинга и целостности данных.

Все визуализации создаются с использованием библиотек matplotlib и seaborn, обеспечивающих высокое качество и научную строгость графиков. Графики сохраняются в векторных и растровых форматах, что позволяет использовать их в публикациях и отчётах. Модуль визуализации интегрирован как в CLI, так и в GUI-версию системы, обеспечивая единообразие результатов независимо от способа взаимодействия.

**4. Тестирование и сборка исполняемого файла**

**4.1. Тестирование программы на различных логах**

Для верификации корректности работы программного комплекса была проведена серия тестов на разнообразных лог-файлах, полученных из реальных экспериментов моделирования в среде Geant4. Тестирование охватывало широкий спектр сценариев: от простых моделей с одной первичной частицей до сложных каскадных процессов с генерацией множества вторичных частиц.

**Методология тестирования:**

**Многоуровневая система проверок включает в себя следующие этапы**:

* + Функциональное тестирование парсинга логов объемом от 4 тыс. до 2.3 млн шагов
  + Верификация корректности распознавания различных типов частиц (γ, e-, proton, neutron, ионы)
  + Проверка точности конвертации единиц измерения (эВ → кэВ → МэВ, мм → см → м)

**Валидация физической корректности** **проверок включает в себя следующие этапы**:

* + Контроль энергетического баланса: Начальная энергия = Energy deposit + Energy leakage
  + Сравнение частот процессов между данными парсинга и итоговой сводкой Geant4
  + Анализ пространственного распределения частиц в детекторе

**Ключевые выводы тестирования:**

1. **Валидация энергетического баланса**: Во всех тестовых случаях расхождение между начальной энергией первичных частиц и суммой (Energy deposit + Energy leakage) не превышало 1%, что подтверждает физическую корректность моделирования и точность парсинга.
2. **Анализ неполных логов**: Тестирование выявило, что большинство логов имеют сокращенный формат, где отсутствуют промежуточные шаги. Это подтверждается значительными расхождениями при использовании Метода 1 (сумма dEStep) и практически идеальными результатами при использовании Метода 2 (энергетический баланс треков).
3. **Масштабируемость**: Программа успешно обрабатывает логи объемом до 2.3 млн шагов, демонстрируя линейную зависимость времени обработки от объема данных.
4. **Универсальность**: Подтверждена корректная работа с различными типами частиц и физическими процессами, включая ядерные реакции, электромагнитные взаимодействия и процессы с тяжелыми ионами.

**4.2. Сборка исполняемого приложения под Windows**

Для обеспечения кроссплатформенной совместимости и удобства распространения была разработана система сборки автономного исполняемого файла для операционной системы Windows.

**Архитектура системы сборки:**

1. **Технологический стек**:
   * **PyInstaller 5.0+** — инструмент для создания standalone приложений
   * **Специализированные batch-скрипты** — автоматизация процесса сборки
   * **Виртуальное окружение** — изоляция зависимостей и воспроизводимость