

Министерство науки и высшего образования Российской Федерации
Федеральное государственное автономное образовательное учреждение
высшего образования
«Национальный исследовательский ядерный университет МИФИ»
(НИЯУ МИФИ)

УДК 53.05, 53.07

ОТЧЁТ
О НАУЧНО-ИССЛЕДОВАТЕЛЬСКОЙ РАБОТЕ
АВТОМАТИЗАЦИЯ ПОДСЧЁТА СИСТЕМАТИЧЕСКИХ
ПОГРЕШНОСТЕЙ АНАЛИЗА В ЗАДАЧЕ ПОИСКА
МНОГОЗАРЯДНЫХ ЧАСТИЦ

Научный руководитель

к.ф.-м.н.

_____ Ю. С. Смирнов

Выполнил

_____ Г. И. Воробьев

Москва 2020

Содержание

Содержание	1
1 Введение	2
1.1 Стандартная модель	2
1.2 МСР	3
1.3 Поиск МСР	4
1.3.1 Моделирование и анализ результатов	4
1.3.2 Критерии отбора событий	5
1.3.3 Эффективность поиска и погрешности на сечения рождения многозарядных частиц	8
2 Программная реализация	10
2.1 Парсинг файлов моделирования	10
2.2 Работа с базами данных SQLite3	12
3 Расчёт систематических погрешностей анализа и вывод таблицы .csv	12
4 Интерфейс пользователя и библиотека click	14
4.1 Структура программы	14
5 Заключение	17
Список использованных источников	19

Цель работы

Изучение концепций многозарядных частиц и методов их поиска в эксперименте ATLAS. Разработка программного обеспечения на языке программирования Python для сбора в единую базу результатов вычисления всех погрешностей на эффективность поиска многозарядных частиц в моделировании; Вычисление по этим данным систематических погрешностей этого анализа; Визуализация этих погрешностей в виде .csv таблицы.

1 Введение

1.1 Стандартная модель

Стандартная модель (СМ) — это теория описывающая фундаментальные частицы и их взаимодействия между собой. Она объединяет электромагнитное, слабое и сильное взаимодействие. На данный момент все её предсказания с высокой точностью подтверждаются экспериментально и успешно объясняет большую часть явлений в физике частиц. Однако множество вопросов, в том числе: описание гравитационного взаимодействия, иерархия частиц, осцилляции нейтрино, тёмная энергия и материя — остаются без ответа. Как следствие, главной задачей современной науки стал поиск "*новой физики*" — пределов применимости СМ. В то же время, разрабатываются теории, основанные на других принципах или дополняющие, поправляющие СМ. Например, основанная на "почти коммутативной" геометрии (см. [1]) АС-модель, модель «бегущего» техницвета WTC (см. [2]) или модель лево-правой симметрии (см. [3]). Эти теории выделяются тем, что предсказывают существование долгоживущих частиц с электрическим зарядом выше элементарного, поиском которых занимается одна из групп в ЦЕРН на эксперименте ATLAS. Подтверждением существования таких частиц станет той самой *новой физикой* вне СМ, а так же сможет внести ясность в проблемы скрытой массы и новых симметрий вне рамок СМ.

масса→	$\approx 2.3 \text{ МэВ}/c^2$	$\approx 1.275 \text{ ГэВ}/c^2$	$\approx 173.07 \text{ ГэВ}/c^2$	0	$\approx 126 \text{ ГэВ}/c^2$
заряд→	2/3	2/3	2/3	0	0
спин→	1/2	1/2	1/2	1	0
	u верхний	c очарованный	t истинный	g глюон	H бозон Хиггса
КВАРКИ	$\approx 4.8 \text{ МэВ}/c^2$	$\approx 95 \text{ МэВ}/c^2$	$\approx 4.18 \text{ ГэВ}/c^2$	0	
	-1/3	-1/3	-1/3	0	
	1/2	1/2	1/2	1	
	d нижний	s странный	b прелестный	γ фотон	
	$0.511 \text{ МэВ}/c^2$	$105.7 \text{ МэВ}/c^2$	$1.777 \text{ ГэВ}/c^2$	$91.2 \text{ ГэВ}/c^2$	
	-1	-1	-1	0	
	1/2	1/2	1/2	1	
	e электрон	μ мюон	τ тау	Z Z бозон	
ЛЕПТОНЫ	$< 2.2 \text{ эВ}/c^2$	$< 0.17 \text{ МэВ}/c^2$	$< 15.5 \text{ МэВ}/c^2$	$80.4 \text{ ГэВ}/c^2$	
	0	0	0	$\neq 1$	
	1/2	1/2	1/2	1	
	ν_e электронное нейтрино	ν_μ мюонное нейтрино	ν_τ тау нейтрино	W W бозон	
					КАЛИБРОВОЧНЫЕ БОЗОНЫ

Рисунок 1.1 – 6 кварков, 6 лептонов и 5 бозонов Стандартной модели

1.2 МСР

Многочарядные частицы (МСР – англ. *multi-charged particles*) – гипотетические частицы, имеющие по модулю электромагнитными заряд $2e$ и выше.

АС-модель, основанная на «почти коммутативной геометрии» предсказывает существование 2-ух тяжёлых частиц с противоположным электромагнитным зарядом. Модуль их электромагнитного заряда теорией не фиксируется, т.е. допустимы значения больше e .

В некоторых вариантах модели техницвета, таких как WTC-модель (модель «бегущего» техницвета) существует 4-ое семейство лептонов - нейтрино ν' и соответствующего ему второй лептон ζ с зарядами $\frac{1-3q}{2}$ и $\frac{-1-3q}{2}$ соответственно.

Модель лево-правой симметрии расширяющая физику бозона Хиггса. Она вводит новый тип слабого взаимодействия, в котором появляется двухзарядный Хиггс-бозон $H^{\pm\pm}$. Для него будут характерны только лептонные

моды распада (в рамках этой модели). Если его лептонный заряд будет равен нулю, то, из-за не сохранения этого заряда в его распаде, Хиггс-бозон был бы долгоживущим.

Эти частицы рассматриваются как стабильные, а так же как зарядно-нейтральные к сильному взаимодействию (оба этих критерия выполняются в описанных ранее моделях).

1.3 Поиск МСР

1.3.1 Моделирование и анализ результатов

Моделирование изучаемых явлений (обычно по методу Монте-Карло или ММК), а так же поведения оборудования в этих процессах - неотъемлемая часть анализа в экспериментальной физике высоких энергий. Для подобного рода исследований существуют различные программы использующие различные алгоритмы ММК для различных задач. В рамках нашей группы использовалась программная оболочка "ATHENA" с генераторами MADGRAPH и PYTHIA. MADGRAPH создаёт пары искомых частиц со спином $1/2$, рождающихся от изначального pp -взаимодействия через кварк-антикварковую аннигиляцию в виртуальный фотон (рис. 1.2) А PYTHIA

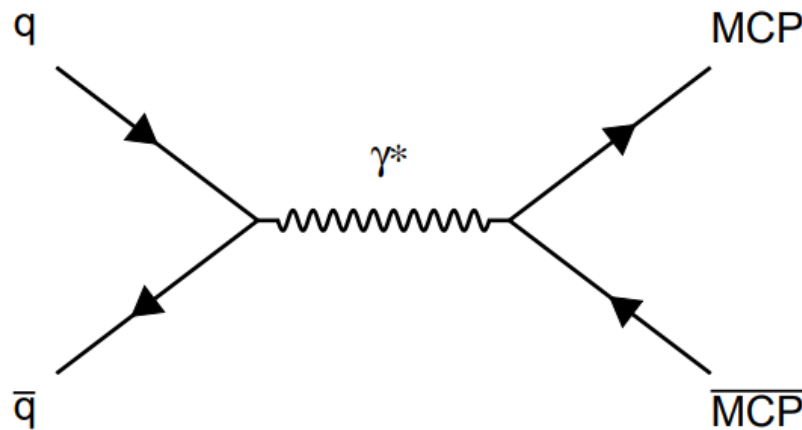


Рисунок 1.2 – Диаграмма Фейнмана рождения многозарядных частиц (" MCP " и " \overline{MCP} ") через кварк-антикварковую аннигиляцию в виртуальный фотон.

описывает финальные состояния всех остальных частиц СМ, рождающихся вместе с искомыми в протон-протонных столкновениях. Далее моделируются отклик самого детектора на эти события (с помощью GEANT4) и связанной с ним электроники. [4]

После этого с каждой подсистемы детектора обрабатываются данные сначала в отдельности, а потом комбинировано, восстанавливая события в исходных состояниях. Аналогично этой итерации проходит анализ экспериментальных данных.

1.3.2 Критерии отбора событий

Имея некоторые ранее полученные представления о новых частицах (из теорий описанных в главе 1.2) можно составить минимальные условия ограничивающие события по некоторым параметрам. Если число соответствующих этим условиям событий будет значительно превосходить ожидаемый фон – частица открыта. В противном случае, когда разница между сигналом и фоном не является статистически значимой или же фон даже преобладает над наблюдаемым сигналом, можно уточнить верхние пределы на сечения рождения этих частиц.

Рассмотрим общие условия на события (обработка событий отсеиванием не интересующих нас данных) в их рабочем порядке:

1. Отсеивание событий во время сбоев в работе детекторов ATLAS (“Good Runs List”- фильтрация или GRL-отбор) – ограничение только экспериментальных данных
2. Наличие в событии хотя бы одного мюона («мюонное фильтрация»)
3. Перед сопоставлением сегментов треков, восстановленных во внутреннем детекторе и в мюонной системе, в единый для каждого сегмента независимо друг от друга выполняются соответствие критериям:
 - а) для сегментов в мюонной системе: Предварительного отбора, которые, в основном, определяют кинематическую область поиска треков и гарантируют их высокое качество с точки зрения достаточного количества хитов в разных детекторах, по которым впоследствии восстанавливается трек. После чего – триггерного отбора, который проверяет, сработал ли в событии хоть один из триггер из комбинации триггеров на одиночный мюон, поперечную энергию и запаздывающий мюон.

б) для сегментов во внутреннем детекторе: условия описанные на Рис. 1.3.

4. Если работа ведётся с результатами моделирования, а не с экспериментальными данными, аналогичным образом такой единый трек сопоставляется с треком сгенерированной многозарядной частицы (“сигнальная truth-частица”), причём на её параметры с самого начала накладываются два ограничения из тех, которые накладываются на сегмент трека, восстановленный внутренним детектором: на поперечный импульс и псевдобыстроту. В итоге мы получаем единый восстановленный трек, принадлежащий сигнальной truth частице.
5. Основной отбор.
6. Финальный отбор и оценка фона.

Параметр	Ограничение
Поперечный импульс	$p_T > 20 \text{ ГэВ}$
Псевдобыстрота	$ \eta < 2.0$
Поперечный прицельный параметр (расстояние между точкой наибольшего приближения трека (перигея) и соответствующей первичной вершиной)	$d_0 < 1.5 \text{ мм}$
Продольный прицельный параметр	$ z_0 \sin \theta < 1.5 \text{ мм}$
Число хитов в SCT	$N_{\text{SCT hits}} \geq 6$
Число хитов в TRT	$N_{\text{TRT hits}} \geq 10$
Величина TRT dE/dx	Определена

Рисунок 1.3 – Список ограничений на параметры сегментов трека во внутреннем детекторе, применяемых до их сопоставления с сегментами, восстановленными мюонной системой.

Основной отбор – это “жесткий” отбор событий с целью максимально снизить вклад от фона минимизируя потери сигнала.

- Для зарядов $\pm 2e$: $S(\text{Pixel } dE/dx) > 10$, где $S = \frac{dE/dx - \mu_{Z^0 \text{ decay muons } dE/dx}}{\sigma_{Z^0 \text{ decay muons } dE/dx}}$, что достаточно хорошо разделяет области сигнала и фона. Вероятность искомой частице иметь $S(\text{Pixel } dE/dx) > 10$, равна $8 \cdot 10^{-3}$, а вероятность неверно идентифицировать мюон как двухзарядную частицу равна $2.5 \cdot 10^{-3}$;

- Для зарядов $2e \leq |q| \leq 6e$: ограничение на долю хитов на треке частицы в TRT, сработавших с превышением верхнего порога по амплитуде – $f^{HT} > 0.4$. Такое условие отбора для поиска частиц с зарядами подавляет фон в 10^3 раз без какой-либо потери сигнала.

Финальный отбор треков производится с помощью переменных значимости потерь в TRT и MDT одновременно. Для оценки фона из экспериментальных данных в 2011 использовался так называемый ABCD-метод (пример использования на Рис. 1.5). В последних работах он используется только при поиске зарядов равных $2e$.

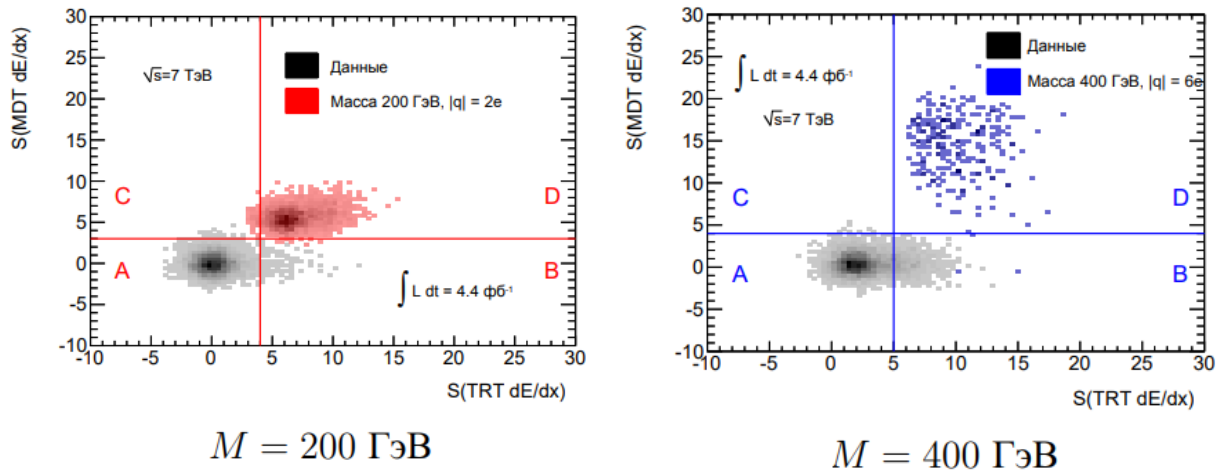


Рисунок 1.4 – Плоскости значимостей потерь в TRT и MDTs после «жёсткого» отбора для случая поиска многозарядных частиц с зарядами $2e$, $3e$. Показаны распределения мюонов в экспериментальных данных 2011 года (периоды F-M) в оттенках серого и многозарядных частиц оттенками синего и красного.

Наконец, ожидаемое число треков в экспериментальных данных $N_{\text{data}}^{\text{expected in D}}$ в каждом сигнальном квадранте может быть вычислено на основании их наблюдаемого количества в остальных областях – $N_{\text{data}}^{\text{observed in A}}$, $N_{\text{data}}^{\text{observed in B}}$, $N_{\text{data}}^{\text{observed in C}}$ – по формуле 1.1:

$$N_{\text{data}}^{\text{expected in D}} = \frac{N_{\text{data}}^{\text{observed in B}} \cdot N_{\text{data}}^{\text{observed in C}}}{N_{\text{data}}^{\text{observed in A}}} \quad (1.1)$$

Для зарядов больше $2e$ эта формула не используется, т.к. $N_{\text{data}}^{\text{observed in C}} = 0$. В этом случае выбирается приближение: $N_{\text{data}}^{\text{expected in D}} = f \cdot N_{\text{data}}^{\text{observed in B}}$, где f вероятность наблюдать событие с частицей $S(\text{MDT } dE/dx) > 4$ с использованием распределения $S(\text{MDT } dE/dx)$ после выбора методом “anti-tight”

(предоставляет много статистики и ортогонален обычному выбору, поэтому нет риска непреднамеренного засвечивания). [5]

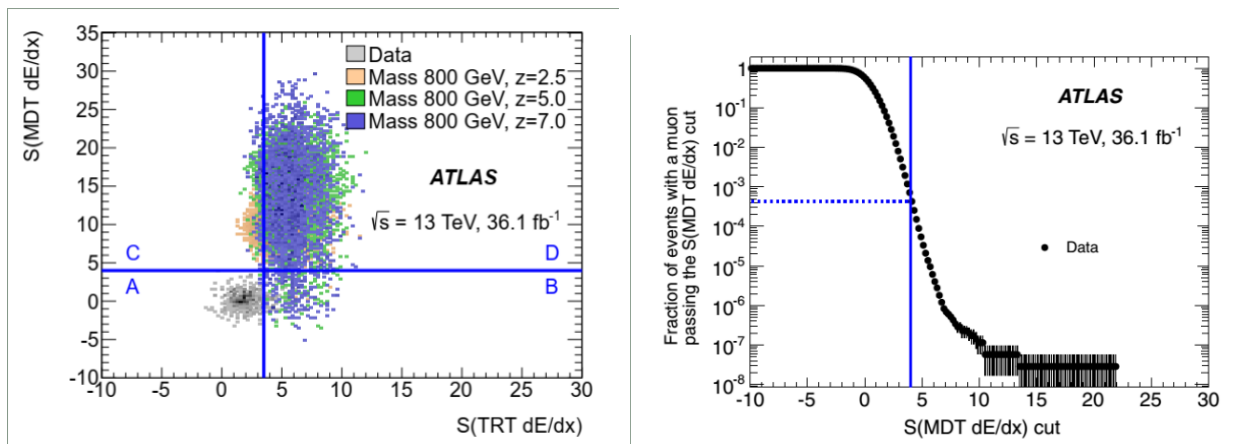


Рисунок 1.5 – Оценка фона при зарядах искомых частиц более $2e$

1.3.3 Эффективность поиска и погрешности на сечения рождения многозарядных частиц

Предел на сечение процесса рождения σ (ожидаемый или наблюдаемый) определяется из отношения:

$$\sigma_{\text{limit}} = \frac{N_{\text{data}}}{\varepsilon \cdot \mathcal{L}}, \quad (1.2)$$

где N_{data} – количество наблюдаемых сигналоподобных событий в экспериментальных данных за вычетом фона на уровне достоверности 95%, ε – эффективность поиска, \mathcal{L} – интегральная светимость проанализированных экспериментальных данных. σ_{limit} также зависит от погрешностей на все эти величины, а именно от:

1. $\sum_i \Delta \varepsilon_i$ – суммарной погрешности (i – один источник погрешности) на эффективность;
2. $\Delta \mathcal{L}$ – погрешности на светимость (известная величина);
3. ΔN – сумме статистической и систематической погрешности на количество сигналоподобных событий в экспериментальных данных из фона.

Для каждого этапа отбора событий существует погрешность, рассчитываемая программой из поиска этих событий в эксп. данных и вычисления эффективности их поиска в моделировании. Итоговые погрешности представляют собой квадратичную сумму всех погрешностей отдельно для каждой изученной массы многозарядных частиц. [4]

Кратко опишем основные:

- систематической погрешности из-за недостаточного совпадения спектров параметров, используемых в анализе, между экспериментальными данными и результатами моделирования (не превышает 2.2%).
- статистической погрешности из-за ограниченной статистики при моделировании сигнальных выборок (может достигать до 30% в случае максимального заряда и минимальной массы);
- систематической глобальной погрешности на эффективность триггерного отбора (1%);
- систематической погрешности на эффективность триггерного отбора, вытекающей из варьирования поправочного коэффициента ρ на $\pm 50\%$ для каждого его значения (до 25%);

Суммарная погрешность $\sum_i \Delta \varepsilon_i$ вычисляются путём применения вариаций, то есть: путём изменения какого-то определённого внутреннего параметра, вычислением новой эффективности поиска, и сравнением этой новой эффективности со старой. Относительная разница и есть погрешность.

2 Программная реализация

На момент начала работы у научной группы имелось две программы:

1. Программа поиска многозарядных частиц в экспериментальных данных и вычисления эффективности их поиска в моделировании (Вычисляет эффективность поиска ε , отдельно каждую $\Delta\varepsilon_i$, ΔN и N)
2. Программа статистического анализа, рассчитывающая σ_{limit} (На входе ε , $\Sigma_i \Delta\varepsilon_i$, \mathcal{L} , $\Delta\mathcal{L}$, ΔN и N)

Соответственно проблемой является именно то, что две программы никак не связаны между собой, из-за чего приходится буквально вручную переводить огромное кол-во данных, а так же рассчитывать $\Sigma_i \Delta\varepsilon_i$ отдельно (программа 1 на выходе имеет только массив из $\Delta\varepsilon_i$, тем временем, как нам нужны суммарные погрешности).

В работе этого семестра решалась часть этой задачи, а именно: получение данных из первой программы и их обработка с целью вычислить $\Sigma_i \Delta\varepsilon_i$. В рамках научной работы было решено изучить вопрос программной реализации чуть глубже и ознакомиться с некоторыми базовыми методами создания UI (англ. User Interface - интерфейс пользователя)

Каждая подзадача выполняется отдельно написанными функциями. Они и будут рассмотрены ниже.

2.1 Парсинг файлов моделирования

Основная функция для работы с файлами и занесение данных из них в единую базу данных называется *makedb*, вызываемая функцией UI *mkdb*. На вход подаются переменные: *path* – путь к файловой системе результатов программы 1, *pathdb* – путь к единой базе данных, *namedb* – её название. Зная структуру файловой системы *path* (Рис. 1.6) программа циклом перебирает каждую папку файловой системы и ищет внутри них файл ".../submit/log-0.out".

Синтаксический анализатор (жарг. **парсер** - англ. *parser*) — это программа или часть программы, выполняющая синтаксический анализ. В этой программе парсинг осуществляется библиотекой "*RE*".

Этот компьютер > Диск (D:) > Download > variationJobs_24Sept2020

Имя	Дата изменения	Тип
submitDir_2020-September-24_18h20m46s843104329ns_M500_ZZ_	08.10.2020 17:24	Папка с файлами
submitDir_2020-September-24_18h20m53s493004780ns_M800_ZZ_	08.10.2020 17:24	Папка с файлами
submitDir_2020-September-24_18h20m57s244354457ns_M1100_ZZ_	08.10.2020 17:24	Папка с файлами
submitDir_2020-September-24_18h21m00s818117547ns_M1400_ZZ_	08.10.2020 17:24	Папка с файлами
submitDir_2020-September-24_18h21m04s371415713ns_M1700_ZZ_	08.10.2020 17:24	Папка с файлами
submitDir_2020-September-24_18h21m07s554601384ns_M2000_ZZ_	08.10.2020 17:24	Папка с файлами
submitDir_2020-September-24_18h21m10s650025941ns_M500_ZZ_MUON_EFF_RECO_STAT_1down	08.10.2020 17:24	Папка с файлами
submitDir_2020-September-24_18h21m14s617648722ns_M800_ZZ_MUON_EFF_RECO_STAT_1down	08.10.2020 17:24	Папка с файлами
submitDir_2020-September-24_18h21m17s699816967ns_M1100_ZZ_MUON_EFF_RECO_STAT_1down	08.10.2020 17:24	Папка с файлами
submitDir_2020-September-24_18h21m20s808325589ns_M1400_ZZ_MUON_EFF_RECO_STAT_1down	08.10.2020 17:24	Папка с файлами
submitDir_2020-September-24_18h21m24s041034327ns_M1700_ZZ_MUON_EFF_RECO_STAT_1down	08.10.2020 17:24	Папка с файлами
submitDir_2020-September-24_18h21m27s216466555ns_M2000_ZZ_MUON_EFF_RECO_STAT_1down	08.10.2020 17:24	Папка с файлами
submitDir_2020-September-24_18h21m30s374887454ns_M500_ZZ_MUON_EFF_RECO_STAT_1up	08.10.2020 17:24	Папка с файлами
submitDir_2020-September-24_18h21m33s517846781ns_M800_ZZ_MUON_EFF_RECO_STAT_1up	08.10.2020 17:24	Папка с файлами
submitDir_2020-September-24_18h21m36s529723719ns_M1100_ZZ_MUON_EFF_RECO_STAT_1up	08.10.2020 17:24	Папка с файлами
submitDir_2020-September-24_18h21m39s521555773ns_M1400_ZZ_MUON_EFF_RECO_STAT_1up	08.10.2020 17:24	Папка с файлами

Рисунок 1.6 – Структура файловой системы результатов программы 1

Каждая папка предварительно *парсится* по названию (т.е. производится поиск ключевых или шаблонных слов, для ориентации в файле, с последующим использованием этой информации для получения данных их файла), чтоб получить информацию о названии проводимой вариации "ZZZ", массе "XXX" в ГэВ, модуле заряда частицы "Y", времени запуска программы "TTT" и специального параметра вариации "UP/DOWN"

```

1 import re
2 #submitDir_2020-September-24_TTT_MXXX_ZY_ZZZ/
3 strRE1 = re.compile("submitDir_(.*)(...)ns_M(*)_Z(*)_(.)__1
   ↪ (.*)") # down\up trig
4 strRE2 = re.compile("submitDir_(.*)(...)ns_M(*)_Z(*)_(.)")
   ↪ #without

```

Нам известна общая структура этого файла, а так же интересующая нас строка:

```

1 AnalysisAlg          INFO      The "ZZZ" variation was
   ↪ applied, so we compare the final efficiencies for MXXXZY:
   ↪ difference is -0.0483276%

```

Здесь нам нужно получить лишь значение разницы ("*difference*").

Все эти параметры сохраняются в переменные и сразу записываются в базу данных.

2.2 Работа с базами данных SQLite3

В качестве хранилища данных использовались методы библиотеки "*SQLite3*" [6]. Она удобна в использовании и достаточно быстродействующая. Основная её особенность, что, по сути, она позволяет внутри кода на питоне использовать язык *SQL* для работы с базами данных. Ниже приведён фрагмент скрипта объявляющего SQL-таблицу из интересующих нас переменных, где "*key*" – параметра вариации "*UP/DOWN*":

```
1      Import SQLite3
2      # CREATE TABLE
3      cursor.execute("""CREATE TABLE AE_DATA
4          (variation_name text, compilation_date text UNIQUE ON
5              ↪ CONFLICT IGNORE, charge real,
              energy integer, difference real, key text)""")
```

После выполнения функции *makedb* создаётся файл формата *.db* в котором хранятся все данные о всех погрешностях, что были найдены в файловой системе и не имеют ошибок. Параллельно этому файлу в логах программы были записаны файлы вариаций, которые не прошли проверку по какому-либо из критериев (ошибка структуры и т.д.).

3 Расчёт систематических погрешностей анализа и вывод таблицы *.csv*

после создания базы данных можно обработать их для дальнейшего использования и визуализации, с помощью функции *csv_writer*. На вход она получает путь и название базы данных, конечный заряд и его шаг (для точной настройки таблицы) и параметр времени, чтоб можно было выбрать данные из определённого промежутка.

Далее таблица по заданному шаблону заполняется сначала в многомерный список. После чего происходит расчёт погрешностей по заданным формулам. Конечный многомерный список записывается в *.csv* файл с помощью библиотеки "*csv*". Записываются две *.csv* таблицы: полная и с только промежуточными погрешностями. А так же, отдельно, записывается *.txt* файл содержащий только конечные значения $\Sigma_i \Delta \varepsilon_i$.

Charge z	Mass [GeV]	Limited agreement of key variables between data and MC																				Result on the data/MC agreement uncertainty
		pT		S(pixel dE/dx)		TRT fHT		N IBL overflow clusters		S(TRT dE/dx)		S(MDT dE/dx)										
		up	down	up	down	up	down	up	down	up	down	up	down	up	down	up	down					
2	50	-3.583536	1.123158	-5.944088	2.348221	0	0	0	0	-13.215064	6.377088	-0.166942	0.195975	14.92816918	0.0							
	200	-0.825082	0.639373	-4.954328	2.865444	0	0	0	0	-11.237221	5.622698	-0.129461	0.133562	12.30930952	0.0							
	400	-0.033567	0.063876	-4.022227	2.638495	0	0	0	0	-8.152743	3.847223	-0.103281	0.065429	9.091769661	0.0							
	600	-0.006165	0.022049	-3.828542	2.547807	0	0	0	0	-7.849395	3.889833	-0.018914	0.051346	8.733490613	0.0							
	800	-0.029244	0.030835	-3.823685	2.312794	0	0	0	0	-6.109968	2.81655	-0.053941	0.054942	7.208068074	0.0							
	1000	-0.008396	0.010064	-3.34353	2.192995	0	0	0	0	-5.631367	2.55492	-0.030049	0.022912	6.549235938	0.0							
	1400	-0.008614	0.027454	-3.997777	2.605354	0	0	0	0	-5.073458	1.976765	-0.000062	0.009509	6.459337517	0.0							
2.5	50	-2.778253	2.841737	0	-0.72813486	17.354694	-0.5354	-0.000008	-27.810623	16.962413	-0.000008	-0.000008	0.021135	6.095214229	0.0							
	200	-1.327841	0.727475	0	-0.51806216	8.871673	-0.77523	-0.000001	-31.468853	11.77872	-0.001775	0.009567	77.99741417	-0.0								
	400	-0.251353	0.194473	0	-0.28998132	4.563843	-0.990806	0	-24.850779	8.954626	-0.007404	0	60.63445357	0.0								
	600	-0.044828	0.029363	0	-0.17203205	3.52989	-1.006114	0	-21.591362	8.077471	-0.007409	0.028973	38.20337428	0.0								
	800	-0.014991	0.008244	0	-0.12871066	2.036966	-0.887471	0	-18.882547	6.842691	-0.007356	0.010265	27.62521112	0.0								
	1000	-0.007584	0.043482	0	-0.913621	1.424676	-1.236347	0	-16.564815	6.543154	-0.029138	0.007318	22.8692557	0.0								
	1400	-0.035324	0.074799	0	-0.7547135	1.172938	-1.066041	0	-15.762473	6.166896	-0.00188	0.009431	18.9577088	0.0								
3	50	-4.388867	2.269722	0	-0.2116006	0.00001	-1.140181	0.000001	-16.237246	5.215184	0.00001	0.00001	17.5087672	0.0								
	200	-1.177954	0.745776	0	-0.153833	0.005834	-1.128053	0	-13.023065	5.918456	-0.000025	0	16.99081532	0.0								
	400	-0.545873	0.330351	0	-0.542301	0	-1.033857	0	-11.219293	4.293764	-0.000248	0.007121	13.17541771	0.0								
	600	-0.063959	0.074791	0	-0.357596	0.014688	-0.753965	0.000001	-10.185913	4.315618	-0.013977	0.000025	11.29307372	0.0								
	800	-0.028444	0.068396	0	-0.278834	0.000001	-1.06563	0.000001	-9.788821	3.563021	-0.006711	0.007277	10.22032048	0.0								
	1000	-0.041201	0.016645	0	-0.104683	0	-0.876828	0	-8.195386	3.729852	-0.007969	0.000111	9.850840737	0.0								
	1400	-0.028788	0.027782	0	-0.090122	0.000001	-0.933632	0.000001	-8.129351	3.655754	-0.02027	0.025573	8.242930219	0.0								
3.5	50	-0.05821	0.054823	0	-0.032848	0	-0.894774	0	-8.058939	3.05957	-0.041649	0.010115	8.183374676	0.0								
	200	-2.28208	2.024585	0	-0.000042	-0.000042	-0.849842	-0.000042	-12.184321	7.54669	-0.000042	-0.000042	12.42528868	-0.0								
	400	-1.037425	1.022273	0	-0.023654	0	-0.897682	0	-9.701615	3.694402	0	0	9.798161905	0.0								
	600	-0.677885	0.437193	0	0.000001	0.000001	-0.74425	0.000001	-7.610768	3.03084	-0.015993	0.00815	7.677075059	0.0								
	800	-0.106405	0.140241	0	-0.029085	-0.000001	-1.047755	-0.000001	-6.463521	3.116296	-0.000188	0.015306	6.549476468	0.0								
	1000	-0.04515	0.044374	0	-0.000124	-0.000001	-1.231938	-0.000001	-6.864568	3.08244	-0.007366	0.008005	6.974381246	0.0								
	1400	-0.033489	0.052489	0	0	0	-0.752106	0	-6.583275	2.890738	-0.000236	0.001014	6.626309187	0.0								
1200	-0.01854	0.136394	0	0	0	-0.822242	0	-5.567062	2.89722	-0.000097	0.048789	5.629320111	0.0									
1400	0.011026	0.047306	0	0	0	0.060771	0	5.55208	2.57508	0.000000	0.000000	5.364100681	0.0									

Рисунок 1.7 – часть таблицы в google sheets

```
PS C:\Users\gree-\Untitled Folder\DSAE> python dsae.py

-----
| Welcome back in DSAE 0.2.7                               (c) 2020, The DSAE Team |
| [Settings set as DEFAULT or like in a previous session] |
-----

Usage: dsae.py [OPTIONS] COMMAND [ARGS]...

  DSAE CLI application

Options:
  --version  Show the version and exit.
  --help     Show this message and exit.

Commands:
  config  Get and set repository or global options.
  mkdb    Makes database from file system with results of modeling
  printdb Create two csv tables and result_column txt with data from db
  readme  Opens README.
PS C:\Users\gree-\Untitled Folder\DSAE>
```

Рисунок 1.8 – интерфейс пользователя

Визуализация происходит отдельно с помощью *google sheets* и нужна больше для визуальной сверки результатов, если появятся ошибки. .csv файл загружается в *google sheets* и автоматически оформляет таблицу (Рис. 1.7).

4 Интерфейс пользователя и библиотека *click*

В рамках усложнения задачи было принято решение изучить основы UI, а именно интерфейсов командной строки, т.к. графический интерфейс может не работать на удалённых машинах вроде *lxfarm* или *lxfplus*. Выбор был сделан на библиотеку *click*^[7], т.к. это один из самых многофункциональных пакетов, с большой практикой применения.

Click работает по принципу: запустите скрипт с параметрами, или, как можно увидеть на скриншоте 1.8, следующим пример ввода в командную строку:

```
1 dsae.py [OPTIONS] COMMAND [ARGS]...
```

4.1 Структура программы

Программа имеет 3-и основные функции + 1 дополнительная.

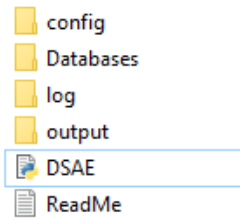


Рисунок 1.9 – файловая структура программы

config Вызывает функцию настроек программы, которая создаёт конфиг файл (используя библиотеку *configparser*), в которой сохраняются основные настройки (путь к файловой системе, путь к единой базе данных, названия и т.д.)

mkdb Описанная ранее, она вызывает функцию *makedb*, создающей базу данных *.db* и сохраняющую в неё данные из файловой системы.

printdb Вызывает функцию *csv_writer*, заполняющую и сохраняющую таблицы

readme путеводитель по программе, описывающий, как работать с функциями описанными выше Рис. 1.11.

Так же в программе ведётся система сбора логов, т.е. файл, в котором записываются программные результаты сессии (ошибки, успехи в ходе работы программы)


```
(C) 2020, The DSAE Team
[Settings set as DEFAULT or like in a previous session]
-----
DSAE - Data Science Analysis Error for multiple charged particle modeling programm

-----
For 29.10.2020 It is just a mini-progect, which create a .db of
modeling error results for multiply-charged particles with
different charge and mass and with special parameter - type
name of variation in modeling. Program asking you the way to
directory with results of modeling, pars names of directories
and in each directory serch log.out file and pars it for result
meaning of difference (error)
-----

Please read below before start work with
==Commands=====
[config]=====
Command config is used to set main settings - 1) name path to
file system with analysis data (inside must be situated
directories with names like:

"submitDir_(data-time)_M(*)_Z(*)_(variation name)",
2) name path to work directory (where program should save .db),
3) name of .db file (where program write data from file sys).

Options:

-d/--default - make command faster. It will ask only
```

Рисунок 1.10 – readme функция

```
PS C:\Users\gree-\Untitled Folder\DSAE> python dsae.py config
-----
| welcome back in DSAE 0.2.7                               (c) 2020, The DSAE Team |
| [Settings set as DEFAULT or like in a previous session] |
-----

[Settings opened]

Enter the path to file-system of modeling, you can use .h for get help :
[DSAE]: .h

----internal-help-----
| Enter the path to file-system of modeling, like:         |
| <D:/Download/variationJobs_24Sept2020> (full path, win), |
| </afs/cern.ch/user/public/data2015_2018_analysis> (linux) |
| also you can use <~/./,../>? like:                       |
| <~user/public/data2015_2018_analysis/>                   |
| <./Database/OCT2020/>                                    |
| [special commands]:                                     |
| .pwd            if you want to see current path          |
| .ls             if want to see another directories in current dir |
| .h              print this help                          |
| ^C(^Z)         means "^" as ctrl.Close app and clear settings.ini |
|-----|
```

Рисунок 1.11 – пример работы с функцией config

5 Заключение

В данной работе был проведён небольшой обзор на проблемы стандартной модели и современной физики частиц, а в частности рассмотрели вопрос существования новых частиц, сходных по свойствам с лептонами, но имеющих заряд больше $1e$. Так же была рассмотрена работа нашей группы в эксперименте ATLAS по поиску таких частиц. В рамках научной работы была разработана программа на языке Python обработки результатов программы поиска таких частиц в экспериментальных данных и вычисления эффективности их поиска в моделировании. Эта программа способна записать результаты вычисления погрешностей от каждого источника в единую базу данных, а после рассчитать суммарные погрешности и вывести их в различных удобных для дальнейшего использования видах. Итоговая таблица содержит 92 строки и 70 столбцов.

В рамках усложнения задачи был разработан интерфейс пользователя для этой программы и получены соответствующие навыки разработки

ПО такого рода.

Дальнейшая работа будет развиваться, с целью полностью автоматизировать расчёты в поиске многозарядных частиц. Разработанная программа послужит объединяющей частью программы поиска и программы дальнейшего анализа, вычисляющей пределы на сечения таких частиц.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. *Khlopov M. Yu. S. C. A.* Composite dark matter with invisible light from almostcommutative geometry. — arXiv:astro-ph/0603187, 2006.
2. *Sannino F. T. K.* Orientifold theory dynamics and symmetry breaking. — Vol. D71. — P. 051901 : Phys.Rev., 2005.
3. *Mohapatra R. N. P. J. C.* Left-Right Gauge Symmetry and an Isoconjugate Model of CP Violation. — Vol. 11 : Phys.Rev.D., 1975.
4. *Смирнов Ю. С.* Результаты поиска долгоживущих многозарядных частиц в эксперименте ATLAS на БАК ЦЕРН. — Диссертация на соискание учёной степени кандидата физико-математических наук, 2017.
5. Search for heavy long-lived multicharged particles in proton-proton collisions at s=13 TeV using the ATLAS detector / M. Aaboud [и др.] // Physical Review D. — 2019. — март. — т. 99, № 5.
6. *Richard Hipp D.* SQLite3. — www.sqlite.org/index.html, 2000.
7. *The Pallets projects.* Click. — click.palletsprojects.com/en/7.x/, 2014.