

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ
ВЫСШЕГО ОБРАЗОВАНИЯ
«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ ЯДЕРНЫЙ УНИВЕРСИТЕТ «МИФИ»
(НИЯУ МИФИ)

ИНСТИТУТ ЯДЕРНОЙ ФИЗИКИ И ТЕХНОЛОГИЙ
КАФЕДРА №40 «ФИЗИКА ЭЛЕМЕНТАРНЫХ ЧАСТИЦ»

УДК 53.05, 53.07

**ПОЯСНИТЕЛЬНАЯ ЗАПИСКА
К БАКАЛАВРСКОЙ ДИПЛОМНОЙ РАБОТЕ
РЕИНТЕРПРЕТАЦИЯ ПОИСКА МНОГОЗАРЯДНЫХ
ЧАСТИЦ И ОБЕСПЕЧЕНИЕ КРОССПЛАТФОРМЕННОСТИ
ЕГО ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ**

Студент

_____ Г. И. Воробьев

Научный руководитель,
к.ф.-м.н.

_____ Ю. С. Смирнов

Москва 2021

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА БАКАЛАВРА

**РЕИНТЕРПРЕТАЦИЯ ПОИСКА МНОГОЗАРЯДНЫХ
ЧАСТИЦ И ОБЕСПЕЧЕНИЕ КРОССПЛАТФОРМЕННОСТИ
ЕГО ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ**

Студент	_____ Г. И. Воробьев
Научный руководитель, к.ф.-м.н.	_____ Ю. С. Смирнов
Рецензент, к.ф.-м.н., доц.	_____ О. В. Булеков
Секретарь ГЭК, к.ф.-м.н.	_____ А. А. Кириллов
Зав. каф. №40, д.ф.-м.н., проф.	_____ М. Д. Скорохватов

Содержание

Введение	4
1 Многочарядные частицы	6
1.1 Модели многочарядных частиц	6
1.2 Поиск многочарядных частиц	6
1.2.1 Моделирование и анализ результатов	6
1.2.2 Критерии отбора событий	7
1.2.3 Эффективность поиска и погрешности на сечения рождения многочарядных частиц	11
2 RECAST	13
2.1 Проблема сохранения и воспроизводимости анализа	13
2.2 Ценность ПО анализа и его реинтерпретация	14
2.3 Обобщение	14
2.3.1 Docker	15
2.3.2 yadage и packtivity	16
2.3.3 ReAna и recast-atlas	16
2.4 Концепция анализа	17
3 Программа, суммирующая погрешности на эффективность поиска	20
3.1 Парсинг файлов моделирования	20
3.2 Работа с базами данных SQLite3	21
3.3 Расчёт систематических погрешностей анализа и вывод таблицы .csv	22
4 Программная реализация сохранения анализа	25
4.1 Упаковка отдельных программ в контейнеры	25
4.2 yadage реализация взаимодействия контейнеров	26
4.3 Реорганизация «инструкций» yadage с помощью высокоуровневых инструментов	27
4.3.1 recast-atlas	28

4.3.2	ReAna	28
5	Тест процедуры реинтерпретации анализа и его результаты .	29
6	Реинтерпретация анализа	30
6.1	High luminosity LHC	30
6.2	Реинтерпретация	31
7	Заключение	34
	Список использованных источников	36

Введение

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 объясняют 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.

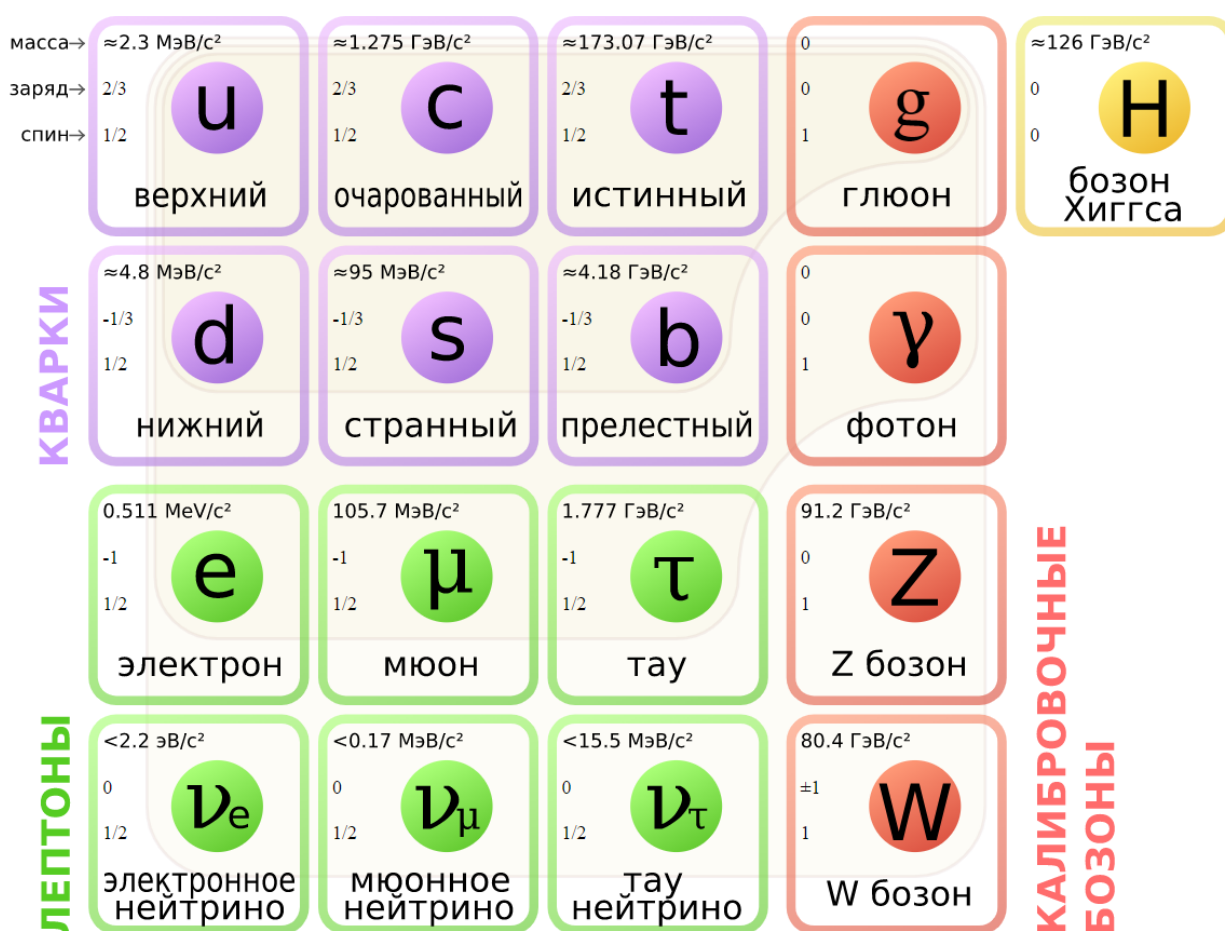


Рисунок 1 — 6 кварков, 6 лептонов и 5 бозонов Стандартной модели

Например, основанная на «почти коммутативной» геометрии (см. [1]) АС-модель, модель «бегущего» техницвета WTC (см. [2]) или модель лево-

правой симметрии (см. [3]) предсказывают существование долгоживущих частиц с электрическим зарядом выше элементарного, поиском которых занимается одна из научно-исследовательских групп в ЦЕРН на эксперименте ATLAS. Подтверждение существования таких частиц станет той самой *новой физикой* вне СМ, а также сможет внести ясность в проблемы скрытой массы и новых симметрий вне рамок СМ.

Целью данной работы является: Изучение метода поиска многозарядных частиц 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 суммарных систематических погрешностей этого анализа; изучение методик фреймворка RECAST для обеспечения кроссплатформенности, сохранения и реинтерпретации анализа; реализация этих методик для программного обеспечения исследуемого поиска.

Полученные результаты: Разработана программа для вычисления суммарных систематических погрешностей этого анализа; разработан рабочий процесс для последовательного запуска ПО анализа, соответствующий критериям RECAST; проведена реинтерпретация анализа для случая более высоких интегральной светимости или эффективности поиска в эксперименте ATLAS; оценено влияние произведённых изменений на нижний предел массы многозарядных частиц.

1 Многозарядные частицы

1.1 Модели многозарядных частиц

Многозарядные частицы — гипотетические частицы, имеющие по модулю электрический заряд $2e$ и выше. Рассмотрим подробнее некоторые теории, допускающие существование таких частиц:

АС-модель (см. [1]), основанная на «почти коммутативной геометрии», предсказывает существование двух тяжёлых частиц (А и С — $SU(2)$ электрослабые синглеты) с противоположным электрическим зарядом и соответствующие им античастицы. Модуль их электрического заряда теорией не фиксируется, т.е. допустимы значения больше e .

В некоторых вариантах модели техницвета, таких как **WTC-модель** (модель «бегущего» техницвета, см. [2]), существует четвёртое семейство лептонов — нейтрино ν' и соответствующий ему лептон ζ с зарядами $\frac{1-3q}{2}$ и $\frac{-1-3q}{2}$ соответственно.

Модель лево-правой симметрии, расширяющая физику бозона Хиггса (см. [3]), 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). При определённых условиях в реакциях распада такой частицы необходимо сохранение лептонного заряда. И если лептонный заряд двухзарядного Хиггс-бозона окажется равным нулю, то такая частица должна быть долгоживущей из-за несохранения лептонного заряда в её распаде. Существование таких частиц расширит наше понимание механизма Хиггса и добавит новые симметрии к Стандартной модели.

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

1.2 Поиск многозарядных частиц

1.2.1 Моделирование и анализ результатов

Моделирование изучаемых явлений (обычно по методу Монте-Карло или ММК), а также отклика электроники в этих процессах — неотъем-

лемая часть анализа в экспериментальной физике высоких энергий. Для подобного рода исследований существуют программы, использующие различные алгоритмы ММК для соответствующих задач. В рамках нашего анализа использовалась программная оболочка «ATHENA» с генераторами MADGRAPH и PYTHIA. MADGRAPH создаёт пары искомых частиц со спином $1/2$, рождающихся либо от изначального pp -взаимодействия через кварк-антикварковую аннигиляцию в виртуальный фотон или виртуальный Z-бозон, либо в результате фотонного синтеза (рис. 1.1).

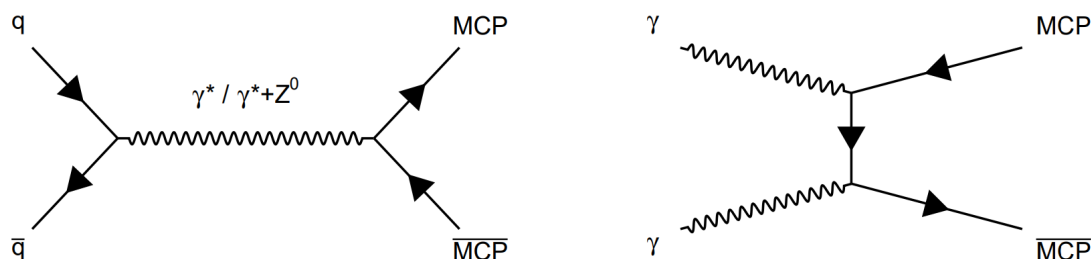


Рисунок 1.1 – Диаграмма Фейнмана рождения многозарядных частиц («MCP» и « \overline{MCP} »). Слева: процесс Дрелл–Яна, справа: процесс фотонного синтеза (t-канал; см. [4; 5])

А PYTHIA проводит моделирование излучения первичной стадии (*parton shower calculation*) и адронизации всех частиц в событии. Далее моделируются отклик самого детектора на эти события (с помощью GEANT4) и связанной с ним электроники [6].

После этого от каждой подсистемы детектора обрабатываются данные сначала в отдельности, а потом комбинированно, восстанавливая события в исходных состояниях.

1.2.2 Критерии отбора событий

Имея некоторые ранее полученные представления о новых частицах (из теорий, описанных в главе 1.1), можно составить минимальные условия, ограничивающие события по некоторым параметрам. Если число соответствующих этим условиям событий в экспериментальных данных будет значительно превосходить ожидаемый фон — частица открыта. В противном случае, когда разница между сигналом и фоном не является статистически значимой или же фон преобладает над наблюдаемым сигналом, можно уточнить верхние пределы на сечения процессов рождения этих частиц.

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

1. Отсеивание событий во время сбоя в работе детекторов ATLAS («Good Runs List» — фильтрация) — ограничение только экспериментальных данных
2. Наличие в событии хотя бы одного мюона («мюонное фильтрация»)
3. После этого события проверяются по критериям триггерного отбора, который проверяет, сработал ли в событии хоть один триггер из комбинации триггеров на одиночный мюон, поперечную энергию и запаздывающий мюон.
4. Перед сопоставлением сегментов треков, восстановленных во *внутреннем детекторе* и в *мюонной системе*, в единый трек проверяются критерии 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 (см. табл. 1.1).
5. Основной отбор.
6. Финальный отбор и оценка фона.

Основной отбор — это «жесткий» отбор событий с целью максимально снизить вклад от фона с минимальными потерями сигнала.

- Для зарядов $\pm 2e$: $S(\text{Pixel } dE/dx) > 13$, 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. dE/dx представляет собой измеренное значение dE/dx исследуемой частицы, $\mu_{Z^0 \text{ decay muons } dE/dx}$ — среднее значение dE/dx мюонов в данных, а $\sigma_{Z^0 \text{ decay muons } dE/dx}$ — среднеквадратичное отклонение распределения мюонов в данных;
- Для зарядов $2e < |q| \leq 7e$: требование наличия по крайней мере одного сработавшего с превышением порога по амплитуде кластера IVL на мюоноподобном треке.

Таблица 1.1 – Список ограничений на параметры сегментов трека во внутреннем детекторе и в мюонной системе

Мюоны		ID трэки	
Параметр	Ограничение	Параметр	Ограничение
Тип	комбинированный	Качество	«LooseMuon» или выше
η, ϕ	$\eta < 2.0$ и вне BMG камер	# TRT-хиты, используемые для расчета TRT dE/dx	≥ 6
Поперечный импульс	$p_T > 50$ ГэВ	Изоляция	нет других ID трэков в пределах $\Delta R < 0.01$
Качество	«среднее» или выше	# пиксельные хиты, используемые для вычисления dE/dx в пиксельном детекторе	≥ 2 (только случай $q = \pm 2$)
Автор	любой, кроме MuGirl	# пиксельных хитов, принадлежащих одновременно как минимум двум трекам	нет (только случай $q = \pm 2$)

Для оценки фона и финального отбора событий из экспериментальных данных используется так называемый ABCD-метод. Его суть состоит в выделении четырёх квадрантов (ABCD-области), в одном из которых находятся преимущественно фон, а в других — сигнал. Пример использования на Рис. 1.2: область с преобладанием фона обозначена как A, а область сигнала обозначена как D (Пример использования метода см. [7]).

Что касается поиска $2e < |q| \leq 7e$ частиц, то применяется тот же самый метод, но используется TRT f^{HT} (доля хитов на сегменте трека в TRT, сработавших с превышением порога по амплитуде с элемента детектора) вдоль оси x вместо значения TRT dE/dx , как показано на рисунке 1.3.

Ожидаемое число событий в экспериментальных данных $N_{data}^{expected}$ в каждом сигнальном квадранте может быть вычислено на основании их наблюдаемого количества в остальных областях ($N_{data}^{observed}$ in A, $N_{data}^{observed}$ in B, $N_{data}^{observed}$ in C) по формуле 1.1.

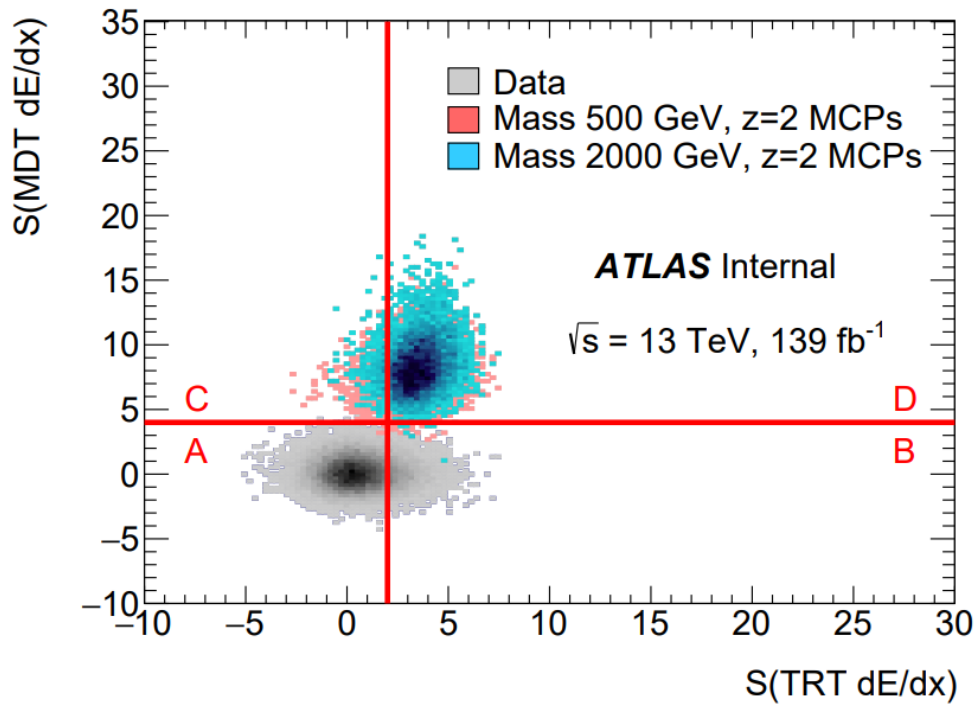


Рисунок 1.2 – Плоскости значимостей потерь в TRT и MDT после «жёсткого» отбора для случая поиска многозарядных частиц с зарядом $2e$. Показаны распределения в экспериментальных данных 2015-2018 года и сигнальные сэмплы (здесь для 500 и 2000 ГэВ) с использованием разных цветов

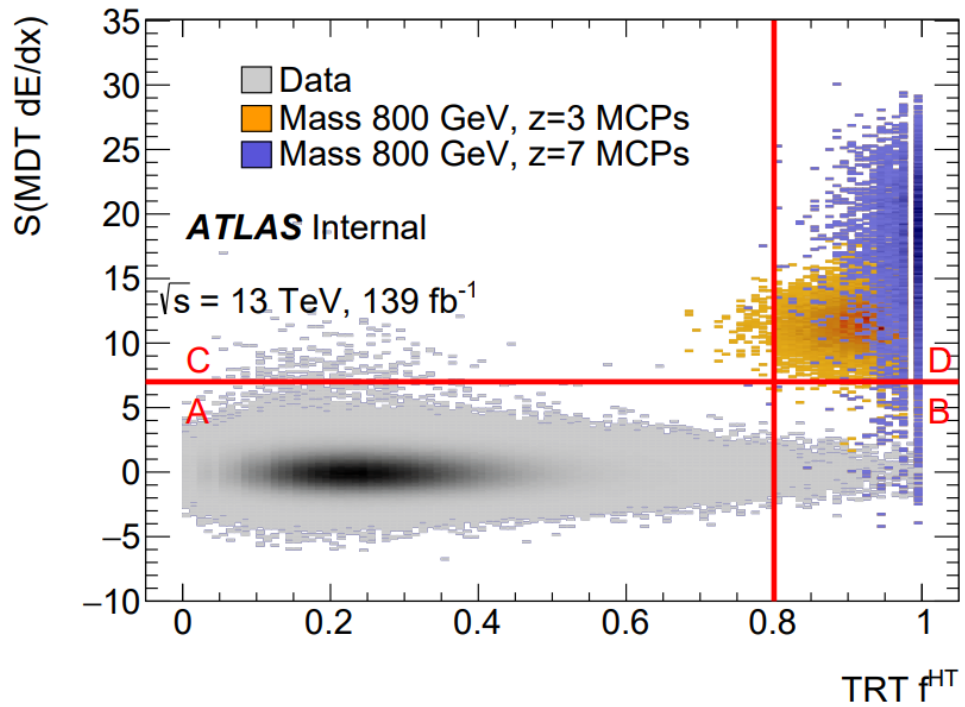


Рисунок 1.3 – Плоскость значимости потерь энергии в MDT и доли хитов на сегменте трека в TRT, сработавших с превышением порога по амплитуде с элемента детектора для случая поиска многозарядных частиц с зарядом $2e < |q| \leq 7e$. Показаны распределения в экспериментальных данных 2015-2018 года и сигнальные сэмплы (здесь $q = 3$ и $q = 7$) с использованием разных цветов

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

В итоге можно описать условия финального отбора следующей таблицей:

Таблица 1.2 – Условия финального отбора в зависимости от заряда

Заряд	Условие
$q = \pm 2e$	$S(\text{TRT } dE/dx) \geq 2.0$ и $S(\text{MDT } dE/dx) \geq 4.0$
$2e < q \leq 7e$	$\text{TRT } f^{\text{HT}} \geq 0.8$ и $S(\text{MDT } dE/dx) \geq 7.0$

1.2.3 Эффективность поиска и погрешности на сечения рождения многозарядных частиц

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

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

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

1. $\Sigma_i \Delta \varepsilon_i$ — суммарной погрешности (i — один источник погрешности) на эффективность;
2. $\Delta \mathcal{L}$ — погрешности на светимость;
3. ΔN — суммы статистической и систематической погрешности на ожидаемое количество фоновых событий.

Суммарные погрешности на эффективность представляют собой квадратичную сумму всех погрешностей от индивидуальных источников от-

дельно для каждой изученной массы и заряда многозарядных частиц [6; 8].

Кратко опишем основные:

- систематическая погрешность из-за недостаточного совпадения спектров параметров, используемых в анализе, между экспериментальными данными и результатами моделирования (может достигать до 9%);
- систематическая погрешность из-за ограниченной статистики при моделировании сигнальных выборок (может достигать до 2.2% в случае максимального заряда и минимальной массы);
- систематическая глобальная погрешность на эффективность триггерного отбора (может достигать до 3.1%);
- погрешность на эффективность восстановления треков (может достигать до 2.9%);
- погрешность на эффективность перевзвешивания pile-up ($\leq 1\%$).

Погрешности $\Delta\varepsilon_i$ в отдельности 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. Суммируя все эти погрешности, получаем таблицу 1.3.

Таблица 1.3 – Суммарные погрешности на эффективность для всех сигнальных сэмплов

		Погрешность, %					
		Заряд					
Масса, ГэВ		2	3	4	5	6	7
		500		2	9	2	3
800		2	6	3	2	4	3
1100		2	5	2	3	3	4
1400		3	4	2	2	3	3
1700		3	4	3	3	3	5
2000		4	3	2	4	5	5

2 RECAST

2.1 Проблема сохранения и воспроизводимости анализа

Современный анализ даже в плане своего программного обеспечения (ПО) — это сложный, комплексный и часто относительно персонализированный (удобный и понятный для разработчика, но не ориентированный на понимание и удобство других пользователей) алгоритм, использующий для своей работы не менее сложные инструменты и утилиты. Незнакомый с каким-либо подобным анализом человек в большинстве случаев не сможет самостоятельно понять его работу и тонкости настройки. Из-за чего возникает потеря важных больших анализов и их программного обеспечения, которое, вероятно, можно было бы ещё использовать.

Рассмотрим эту проблему чуть глубже. Действительно ли после разработки анализа остаётся так много трудностей в его использовании? Выделим основные тонкости:

1. Кроссплатформенность (возможность запустить анализ на разных операционных системах. Например, немногие программы могут работать на ОС Windows и Linux одновременно);
2. Программные компоненты и переменные среды, необходимые для установки и запуска частей анализа. (Нужно, чтобы все вспомогательные инструменты были установлены, настроены, а переменные среды соответствовали тем, что нужны для работы анализа);
3. Формат подачи данных на вход программам из одной в другую;
4. Действия (и их очерёдность) при запуске компонентов анализа;
5. Программное обеспечение имеет свойство устаревать: если его никто не поддерживает, через несколько лет теряется возможность запускать анализ даже вручную самому разработчику.

Все эти нюансы и многие другие, которые могут возникнуть у группы, незнакомой с устройством ПО анализа, отнимают очень много времени и сил. Для решения этой проблемы в ATLAS разработали фреймворк *RECAST*. Это ряд мер, которые разработчики проводят для сохранения и

удобства воспроизводимости своих анализов в дальнейшем. К сожалению, не все проблемы решаются только с помощью RECAST. Много нюансов продолжают скрываться в коде анализа, и тут уже всё зависит от «свободы», которую вложил автор в процессе разработки.

2.2 Ценность ПО анализа и его реинтерпретация

ПО, разработанное для анализа — это большой труд, на который было потрачено много времени. А если ПО написано достаточно качественно, то, вообще говоря, даже после окончания исследований в этой области будут причины обратиться к этому ПО для проверки новых теорий или перепроверки старых данных.

Конкретно ПО нашего анализа может быть использовано для:

- оценки изменения пределов на массу при изменении параметров поиска, например при увеличении его эффективности на 10% или светимости (см. подробнее главу 6);
- тестирования других моделей с другими сечениями процессов рождения искомым частиц;
- вычисления сечения процессов рождения многозарядных частиц с массами, отличными от тех, которые используются в оригинальном анализе.

Для всего этого нужно иметь возможность поменять параметры анализа и перезапустить всю цепочку, независимо от того, возможно ли установить на конкретный компьютер всё необходимое для этого ПО и запустить его в установленной на этом компьютере ОС.

2.3 Обобщение

RECAST [9] (*Request Efficiency Computation for Alternative Signal Theories*) — это фреймворк (каркас или структура инструментов и методов), позволяющий упростить сохранение и повторную интерпретацию анализа ATLAS, а также сделать его кроссплатформенным. С июля 2018 года были добавлены условия для публикации статей группы Exotics, согласно

которым необходимо подготовить анализ для его повторного использования в будущем. Для решения этой задачи в ATLAS был разработан этот фреймворк. Он основан на оперировании над программным обеспечением Docker и веб-инструментом GitLab.

RECAST фреймворк включает в себя несколько инструментов — например, yadage, packtivity, recast-cli и ReAna. Концепт фреймворка заключается в следующем:

1. Отдельные части анализа упаковываются вместе с необходимым программным обеспечением и образом операционной системы в Docker контейнер с помощью GitLab CI/CD pipeline;
2. Программы выполняются внутри контейнера посредством инструмента packtivity (или сразу нескольких контейнеров посредством yadage), который читает написанные для него «инструкции» или «сценарии», описанные в файлах *.yaml*;
3. Используются более высокоуровневые инструменты recast-cli и ReAna для упрощения взаимодействия с инструкциями файлов *.yaml* и контейнерами.

Существуют и альтернативные методы и инструменты, но в данной работе будет рассматриваться именно такой концепт. Подробнее рассмотрим эти инструменты ниже.

2.3.1 Docker



Рисунок 2.1 – Логотип Docker

Введём несколько определений:

Контейнер — это изолированный от других процесс на хост-машине.

Образ контейнера — это настраиваемая файловая система для процесса, который мы именовали контейнером. Соответственно он содержит в

себе всё необходимое для запуска приложения внутри контейнера: все зависимости, конфигурацию, скрипты, двоичные файлы и т.д. Образ также содержит другую конфигурацию для контейнера, такую как переменные среды, команда по умолчанию для запуска, и другие метаданные.

Docker [10] — это программное обеспечение, которое изолирует процессы (как было отмечено ранее, их называют контейнерами), а также позволяет удобно оперировать (сохранять, изменять, запускать) этими контейнерами и их образами. Таким образом, Docker становится центральной фигурой во всём процессе реализации RECAST для анализов. И если на компьютере можно установить Docker, то реализация интерпретации анализа проблем не вызовет.

2.3.2 yadage и packtivity

yadage [11] (*yaml based adage*) — это:

1. синтаксис для описания рабочих процессов, состоящих из контейнерных шагов;
2. механизм выполнения для фактического запуска этих рабочих процессов.

Язык **yadage** сам по себе является примером декларативного языка программирования и основан на парадигме программирования, в которой задаётся спецификация решения задачи, то есть описывается, что представляет собой проблема и ожидаемый результат. Упрощая: с помощью языка **yadage** пишутся «инструкции» или «сценарии» для работы с Docker-контейнерами и их содержимым посредством ввода команд в командной строке.

packtivity — упрощённая версия **yadage**, запускающая лишь один шаг (работает с одним контейнером).

2.3.3 ReAna и recast-atlas

recast-atlas или **recast-cli** — надстройка для организации своего рабочего процесса, взаимодействия с **yadage** и создания модульных тестов.

ReAna [12] — это аналог `recast-cli`, работающий на внешних серверах (Kubernetes, HTCondor и Slurm). Т.е. когда необходимо произвести большой трудоёмкий анализ, то с помощью ReAna можно воспроизвести его на этих серверах. А также, в отличие от `recast-cli`, он поддерживает несколько языков описания рабочих процессов (CWL, yadage и serial).

2.4 Концепция анализа

Наш анализ построен на трёх программах:

1. Программа поиска многозарядных частиц в экспериментальных данных и вычисления эффективности их поиска в моделировании. Разрабатывалась для работы только в *Linux*, а конкретнее на *CENTOS7*.

На вход ей подаются:

- (a) Сигнальные сэмплы,
- (b) Сэмплы с экспериментальными данными.

На выходе из программы мы получаем:

- (a) Гистограммы со свойствами отобранных частиц/событий,
 - (b) Ожидаемое количество фоновых событий из процессов СМ с погрешностью ($N_{\text{фон}} \pm \Delta N_{\text{фон}}$),
 - (c) Итоговые эффективности поиска и все погрешности на них для каждой вариации ($\varepsilon_i, \Delta\varepsilon_j$),
 - (d) Утечки сигнала (доли событий с многозарядными частицами, прошедшими основной отбор, но не попавшими в область D на ABCD-плоскости),
 - (e) Наблюдаемое количество сигналоподобных событий (N).
2. Программа, суммирующая погрешности на эффективность поиска. Реализована на python (см. главу 3) и работает без проблем почти в любой системе, где установлен язык Python.

На вход она берёт:

- Все эффективности поиска и погрешности на них для каждой вариации ($\varepsilon_i \pm \Delta\varepsilon_{ij}$).

На выходе из программы мы получаем:

- Суммарную систематическую погрешность на эффективность поиска многозарядных частиц конкретной массы и заряда, равную:
$$\sqrt{\sum_{j=1}^M \left(\frac{\varepsilon_{\text{orig}} - \varepsilon_j}{\varepsilon_{\text{orig}}} \right)^2}$$

3. Программа статистического анализа для вычисления пределов на сечения процессов рождения и на массу многозарядных частиц.

На вход она берёт:

- Итоговые эффективности поиска (ε_i),
- Ожидаемое количество фоновых событий из процессов СМ с погрешностью ($N_{\text{фон}} \pm \Delta N_{\text{фон}}$),
- Просуммированные погрешности на эффективность ($\Delta\varepsilon_i$),
- Утечки сигнала
- Наблюдаемое количество сигналоподобных событий (N)
- Теоретические сечения процессов рождения многозарядных частиц
- Светимость со своей погрешностью

На выходе из программы мы получаем:

- Ожидаемые и наблюдаемые верхние пределы на сечение процессов рождения на уровне достоверности 95% для каждого сигнального сэмпла
- Полосы $\pm 1\sigma$ и $\pm 2\sigma$ вокруг ожидаемых пределов
- Наблюдаемые и ожидаемые нижние пределы на массу на уровне достоверности 95% (один предел на заряд).

Основной задачей данной работы является реализация автоматизации работы этих трёх программ (передача данных, поочерёдный запуск, а

также реализация кроссплатформенности) с помощью фреймворка RECAST. Схема хода анализа указана на рис.2.2.



Рисунок 2.2 – Требуемый ход анализа

3 Программа, суммирующая погрешности на эффективность поиска

На момент начала этой работы у научной группы имелось только две программы (подробнее см. главу 2.4):

А Программа поиска многозарядных частиц в экспериментальных данных и вычисления эффективности их поиска в моделировании

В Программа статистического анализа, рассчитывающая σ_{limit}

Соответственно проблемой являлось именно то, что две программы никак не связаны между собой, из-за чего приходилось буквально вручную переводить огромное количество данных, а также рассчитывать $\Sigma_j \Delta \varepsilon_j$ отдельно (программа «А» на выходе имеет только массив из $\Delta \varepsilon_{ij}$, тем временем как нам нужны суммарные погрешности).

Подзадачей этой работы была разработка вспомогательной программы (программа 2, см. главу 2.4), которая получает данные из первой программы и обрабатывает их с целью вычислить $\Sigma_j \Delta \varepsilon_j$. Программа представляет из себя набор описанных далее в этой главе функций.

3.1 Парсинг файлов моделирования

Основная функция для работы с файлами и занесение данных из них в единую базу данных называется *makedb*. 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. 3.1), 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*».

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

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 3.1 – Структура файловой системы результатов программы 1

из этого 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 также 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.

3.2 Работа с базами данных SQLite3

В качестве хранилища данных использовались методы библиотеки «*SQLite3*» [13]. Они удобны в использовании и достаточно быстродействен-

ны. Основная особенность этой библиотеки в том, что, по сути, её методы позволяют внутри кода на *Python* использовать язык *SQL* для работы с базами данных. Ниже приведён фрагмент скрипта, объявляющего SQL-таблицу из интересующих нас переменных, где «*key*» — параметр 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.).

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

После создания базы данных они обрабатываются 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 (Столбцы — названия вариаций, колонки — разделения по массе и заряду. Вариации чётко заданы шаблоном, массы выбираются все, что программа смогла найти в базе данных, а заряд имеет гибкий шаблон вида максимальный, минимальный заряд и шаг) 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 таблица, где только промежуточные и конечные суммарные погрешности. А также отдельно записывается *.txt* файл, содержащий только конечные значения $\Sigma_i \Delta \varepsilon_i$. Именно этот файл *.txt* бу-

дет использоваться для передачи данных в последнюю программу, когда мы будем реализовывать сохранение всего анализа.

Визуализацию можно провести отдельно с помощью *google sheets*. Это нужно для визуальной сверки результатов, если появятся ошибки. Файл .csv загружается в *google sheets* и автоматически оформляет таблицу (Рис. 3.2).

Код программы можно посмотреть по ссылке в источнике [14].

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			
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	
	1200	-0.008614	0.027454	-3.997777	2.605354	0	0	0	0	-5.073458	1.976765	-0.000062	0.009509	6.459337517	0.0	
1400	-0.071884	0.023678	-3.702994	3.528936	0	0	0	0	-4.840256	1.975056	-0.078908	0.021135	6.095214229	0.0		
2.5	50	-2.778253	2.841737	0	-0.72813486	17.354694	-0.5354	-0.000008	-0.5354	-27.810623	16.962413	-0.000008	-0.000008	77.99741417	-0.0	
	200	-1.327841	0.727475	0	-0.51806216	8.871673	-0.77523	-0.000001	-0.77523	-31.468853	11.778772	-0.001775	0.009567	60.63445357	0.0	
	400	-0.251353	0.194473	0	-0.28998132	4.563843	-0.990806	0	-0.990806	-24.850779	8.954626	-0.007404	0	38.20337428	0.0	
	600	-0.044828	0.029363	0	-0.17203205	3.52989	-1.006114	0	-1.006114	-21.591362	8.077471	-0.007409	0.028973	27.62521112	0.0	
	800	-0.014991	0.008244	0	-0.12871066	2.036966	-0.887471	0	-0.887471	-18.882547	6.842691	-0.007356	0.010265	22.8692557	0.0	
	1000	-0.007584	0.043482	0	-9.13621	1.424676	-1.236347	0	-1.236347	-16.564815	6.543154	-0.029138	0.007318	18.9577088	0.0	
	1200	-0.035324	0.074799	0	-7.547135	1.172938	-1.066041	0	-1.066041	-15.762473	6.166896	-0.00188	0.009431	17.5087672	0.0	
1400	-0.018373	0.049229	0	-6.838255	0.794818	-0.944329	0	-0.944329	-14.781535	6.062048	-0.018876	0.020222	16.31410733	0.0		
3	50	-4.388867	2.269722	0	-2.116006	0.00001	-1.140181	0.00001	-1.140181	-16.237246	5.215184	0.00001	0.00001	16.99081532	0.0	
	200	-1.177954	0.745776	0	-1.153833	0.005834	-1.128053	0	-1.128053	-13.023065	5.918456	-0.000025	0	13.17541771	0.0	
	400	-0.545873	0.330351	0	-0.542301	0	-1.033857	0	-1.033857	-11.219293	4.293764	-0.000248	0.007121	11.29307372	0.0	
	600	-0.063959	0.074791	0	-0.357596	0.014688	-0.753965	0.000001	-0.753965	-10.185913	4.315618	-0.013977	0.000025	10.22032048	0.0	
	800	-0.028444	0.068396	0	-0.278834	0.000001	-1.06563	0.000001	-1.06563	-9.788821	3.563021	-0.006711	0.007277	9.850840737	0.0	
	1000	-0.041201	0.016645	0	-0.104683	0	-0.876828	0	-0.876828	-8.195386	3.729852	-0.007969	0.000111	8.242930219	0.0	
	1200	-0.028788	0.027782	0	-0.090122	0.000001	-0.933632	0.000001	-0.933632	-8.129351	3.655754	-0.02027	0.025573	8.183374676	0.0	
1400	-0.05821	0.054823	0	-0.032848	0	-0.894774	0	-0.894774	-8.058939	3.05957	-0.041649	0.010115	8.108444571	0.0		
3.5	50	-2.28208	2.024585	0	-0.000042	-0.000042	-0.849842	-0.000042	-0.849842	-12.184321	7.54669	-0.000042	-0.000042	12.42528868	-0.0	
	200	-1.037425	1.022273	0	-0.023654	0	-0.897682	0	-0.897682	-9.701615	3.694402	0	0	9.7998161905	0.0	
	400	-0.677885	0.437193	0	0.000001	0.000001	-0.74425	0.000001	-0.74425	-7.610768	3.03084	-0.015993	0.00815	7.677075059	0.0	
	600	-0.106405	0.140241	0	-0.029085	-0.000001	-1.047755	-0.000001	-1.047755	-6.463521	3.116296	-0.000188	0.015306	6.549476468	0.0	
	800	-0.04515	0.044374	0	-0.000124	-0.000001	-1.231908	-0.000001	-1.231908	-6.864568	3.08244	-0.000366	0.008005	6.974381246	0.0	
	1000	-0.033489	0.052489	0	0	0	-0.752136	0	-0.752136	-2.890738	2.890738	-0.000236	0.000104	6.626309187	0.0	
	1200	-0.01854	0.136394	0	0	0	-0.822242	0	-0.822242	-5.567062	2.89722	-0.000097	0.048789	5.629320111	0.0	

Рисунок 3.2 – часть таблицы в google sheets

4 Программная реализация сохранения анализа

Коллаборация рекомендует реализовывать обеспечение сохранности и кроссплатформенность своего анализа в следующей последовательности:

1. Создать образы Docker для каждой из программных сред, используемых в анализе.
2. Подготовить «сценарии» (steps.yml) для отдельных этапов анализа.
3. Подготовить описание рабочего процесса (workflow.yml) полной цепочки анализа.

4.1 Упаковка отдельных программ в контейнеры

Все три программы были размещены в отдельных репозиториях (см. [15]). Чтобы создать образ каждой программы, нужно написать два файла: «.gitlab-ci.yml» и «Dockerfile». Далее приведем пример первого файла:

```
1 build_image:                                     #.gitlab-ci.yml
2   tags:
3     - docker-image-build
4   script:
5     - ignore
```

Содержание второго файла зависит от необходимых программ внутри контейнера для исправной работы упаковываемой программы (python в образе atlasamglab/stats-base:root6.20.06 предустановлен, поэтому указывается только оболочка терминала zsh, редактор текста nano и библиотека tqdm для python). RUN adduser — используется для уменьшения прав пользователя (нужно для ReAna, во избежание ошибок перезаписи системных файлов).

```
1 FROM atlasamglab/stats-base:root6.20.06        #Dockerfile
2 ADD . /MCPLimitSetting/
3 WORKDIR /MCPLimitSetting/
4 RUN apt-get update && \
5     apt-get install -y \
```

```

6         zsh \
7         nano && \
8     pip3 install tqdm
9     RUN adduser mcp -u 1000 --disabled-password --gecos "" &&
    ↪ usermod -a -G 0 mcp && chown -R mcp /MCPLimitSetting/
10    USER mcp

```

GitLab автоматически распознаёт эти файлы и создаёт docker образ. Далее, получив на странице репозитория название образа, его можно скачивать и пользоваться им на своём устройстве.

Для каждой программы были созданы тестовые данные для проверки работоспособности этих образов. Все три образа успешно прошли проверку с этими данными, в том числе и на операционной системе Windows.

4.2 yadage реализация взаимодействия контейнеров

Язык yadage позволяет запрограммировать отдельно взаимодействие программы с консолью внутри отдельных образов с помощью файла *steps.yml* (называют эту стадию разработки «инструкций» — написание **шагов** рабочего процесса) и отдельно настроить обмен данными между образами (шагами) и очередностью их запуска в *workflow.yml*.

Опишем вкратце общую структуру этих двух инструкций. Пример *steps.yml*:

```

1 eventSelectionStage: #steps.yml
2   process:
3     process_type: interpolated-script-cmd
4     script: |
5       #...
6   publisher:
7     publisher_type: interpolated-pub
8     publish:
9       outputFolderFromTheFirstStage: '{
    ↪ folderWithAllTheSubmitDirFolders}'
10  environment:
11    environment_type: docker-encapsulated
12    image: gitlab-registry.cern.ch/atlas-phys/exot/ueh/
    ↪ multichargedparticles/2015-2018/analysiscode
13  resources:
14    - GRIDProxy

```

Первая строка — название шага. Со строки 4 описываются команды, которые надо запустить внутри контейнера (запуск программы, операции над файлами и т.д.). В *publish* (строка 8) указываются глобальные переменные. В них обычно и сохраняют результаты работы программы. В *environment* записываются названия образов (*GRIDProxy* позволяет реализовать авторизацию внутри контейнера). Пример `workflow.yml`:

```
1 stages:                #workflow.yml
2   - name: eventSelectionStage
3     dependencies: [init]
4     scheduler:
5       scheduler_type: singlestep-stage
6       parameters:
7         mass: {step: init, output: mass}
8         charge: {step: init, output: charge}
9         folderWithAllTheSubmitDirFolders: '{workdir}'
10        #...
11        step: {$ref: 'steps.yml#/eventSelectionStage'}
```

Здесь перечисляются шаги по приведённому шаблону. Вторая строка — название шага, 3 — какие шаги должны быть уже закончены к моменту начала этого, 6 — переменные, 11 — ссылка на шаг внутри *steps.yml*.

4.3 Реорганизация «инструкций» *yadage* с помощью высокоуровневых инструментов

Для удобства использования написанных инструкций используются оболочки *recast-atlas* или *ReAna-client*. Они позволяют удобно сохранить входные параметры и упрощают запуск анализа, частично или полностью решают проблему безопасности авторизации.

Авторизация в подобного рода программах необходима для доступа к данным, хранящимся в частных директориях на серверах EOS (чтобы не хранить всё в репозитории), а также для доступа к самим образам на *dockerhub*. Без оболочек *recast-atlas* или *ReAna-client* авторизацию реализовать можно, но это будет менее удобно и небезопасно.

4.3.1 recast-atlas

Для работы этой оболочки нужно установить `recast-atlas` или воспользоваться Docker-образом, где `recast-atlas` уже установлен. Реализуется в проекте добавлением файла `recast.yml` следующего содержания:

```
1 name: ana-exot-2018-54/recast
2 metadata:
3   author: 'Grigory Vorobyev'
4   input_requirements: 'Input mass and charge'
5   short_description: 'Try to recast for the Exotics UEH multi-
   ↪ charged particle search'
6 spec:
7   workflow: workflow.yml
8 example_inputs:
9   default:
10    dataopts:
11     initdir: '/workflow'
12    initdata:
13     mass: 500
14     charge : 7
```

После добавления анализа в каталог (Рис.4.1) — можно запускать анализ на своём устройстве командой:

```
1 recast run ana-exot-2018-54/recast --tag mpc
```

По завершению работы программы будет создан каталог `recast-mpc` со всеми итоговыми данными. Минус такого метода в ограниченности вычисли-

```
/workflow # recast catalogue ls
NAME                               DESCRIPTION                               EXAMPLES
ana-exot-2018-54/recast            Try to recast for the Exotics UEH multi-charged particle search default
atlas/atlas-conf-2018-041         ATLAS MBJ                                 default
examples/checkmate1               CheckMate Tutorial Example (Herwig + CM1) default
examples/checkmate2               CheckMate Tutorial Example (Herwig + CM2) default
examples/rome                      Example from ATLAS Exotics Rome Workshop 2018 default,newsignal
testing/busyboxtest                Simple, lightweight Functionality Test    default
/workflow # _
```

Рисунок 4.1 – Каталог доступных анализов в `recast-atlas`. Первая строка в списке «`ana-exot-2018-54/recast`» — наш анализ

тельной мощности личного устройства.

4.3.2 ReAna

Отличие `ReAna` от `recast-atlas` в том, что соответствующая задача запускается на удалённых серверах с большими вычислительными мощно-

стями. Реализуется анализ в этой оболочке созданием файла *reana.yaml*:

```
1 version: 0.6.0
2 inputs:
3   files:
4     - reana.yaml
5     - workflow.yaml
6   parameters:
7     mass: 500 # [GeV]
8     charge: 7 # [elementary charges]
9 workflow:
10  type: yadage
11  file: workflow.yaml
12 outputs:
13  files:
14    - limitCalculationStage/results.txt # this is the final
      ↪ file with the limits on the PoI (number of events in
      ↪ the SR)
```

5 Тест процедуры реинтерпретации анализа и его результаты

Разработанные алгоритмы анализа на оболочках recast-atlas (в Windows с помощью Docker^а) и ReAna-client были протестированы и запущены. Необходимые файлы и инструкции были сохранены в репозиторий (см. [16]).

Опытным путём было выяснено, что recast-atlas не работает на ОС Windows, т.к. пытается использовать функции ОС Linux. Для решения этой проблемы снова был использован Docker. Анализ в оболочке recast-atlas был запущен в образе recast/recastatlas:v0.1.0.

Анализ был протестирован для вычисления верхнего предела на сечение процесса рождения многозарядных частиц массой 500 ГэВ и элементарным зарядом $\pm 7e$ при статистике, примерно в 6 раз меньше той (потому что такая статистика соответствует минимально возможному количеству файлов (а именно одному), поданных на вход первой программе), которая используется в анализе, выполненном традиционными средствами. Этот предел оказался равен:

$$\sigma_{limit} = \frac{N_{data}}{\varepsilon \mathcal{L}} = \frac{3.31917}{(5.8/100 \cdot 139)} = 0.4117 \text{ фб}, \quad (1.3)$$

где $N_{data} = 3.31917$ — предел на число наблюдаемых сигналоподобных событий в экспериментальных данных за вычетом фона на уровне значимости 95% в предположении отсутствия сигнала в этих данных, $\varepsilon = 5.8\%$ — результат первого шага программы (эффективность поиска), $\mathcal{L} = 139 \text{ фб}^{-1}$ — интегральная светимость проанализированных экспериментальных данных.

Результат (0.41 фб) не совпадает, но близок к тому, который получился в традиционном анализе (0.39 фб), что объясняется малой статистикой. Поскольку тест был направлен на проверку работоспособности программы в целом, результат получился вполне удовлетворительным.

6 Реинтерпретация анализа

По анализу подготовлено и соответствует критериям коллаборации. Теперь мы можем приступить к заключительной части этой работы — реинтерпретации.

6.1 High luminosity LHC

Периодически LHC (Большой адронный коллайдер) закрывается на обновление оборудования и другие технические работы. В текущий момент идёт очередной перерыв в работе, после которого начнётся фаза «Run-3», а после «Run-4» «high luminosity LHC» (HL-LHC). Суть обновления в повышении эффективности и работоспособности оборудования (в основном имеются ввиду составляющие «Малого колеса» — «The New Small Wheel upgrade») в условиях шумов до 15 кГц/см^2 , которые возникнут, при планируемых значениях светимости (от 2 до $7 \cdot 10^{34} \text{ см}^2 \text{ сек}^{-1}$) [17; 18].

Поэтому, как было описано в главе 2.2, мы можем провести некоторые предсказания об изменениях пределов масс многозарядных частиц в связи с обновлением LHC в целом и экспериментом ATLAS в частности.

6.2 Реинтерпретация

Процедура реинтерпретации пределов на массу частиц с зарядом $\pm 7e$ была также протестирована на Windows. Рассматривались два случая:

1. увеличение интегральной светимости со 139 fb^{-1} до 300 fb^{-1} , что соответствует интегральной светимости, которую, как ожидается, наберёт LHC за Run-3 (2022-2024 годы);
2. увеличение эффективность поиска таких частиц на 10% (второе изменение ничем не оправдано и не ожидается, но тем не менее представляет собой академический интерес).

Напомним, что в текущем анализе график зависимости теоретически предсказанных сечений и экспериментально полученных верхних пределов на сечение процессов рождения частиц с зарядом $\pm 7e$ от их массы выглядит следующим образом:

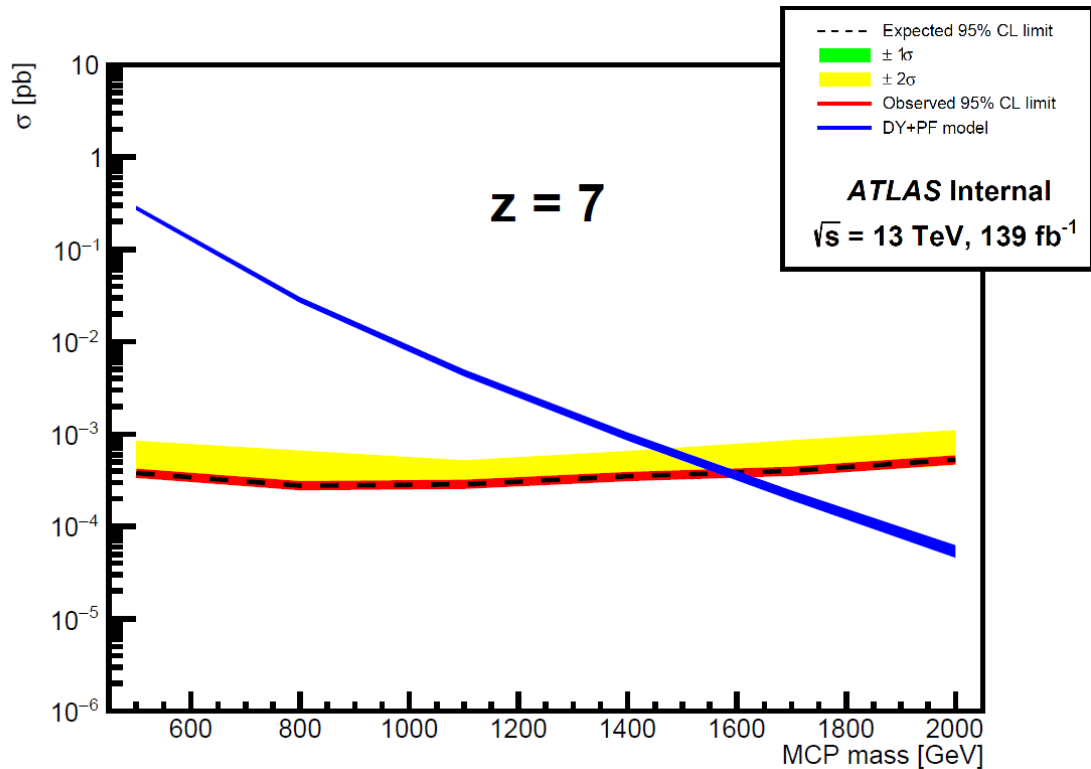


Рисунок 6.1 – Верхние пределы сечения рождения многозарядных частиц в зависимости от их массы при светимости в 139 fb^{-1} . Пунктирная черная линия показывает медиану ожидаемого предела, зеленая и желтая полосы — это полосы неопределенности $\pm 1\sigma$ и $\pm 2\sigma$ вокруг медианы, сплошной красной линией обозначается наблюдаемый предел сечения, синей линией обозначается теоретическое предсказание по модели Дрелл-Яна и фотонного синтеза

Согласно этому графику (рис. 6.1), нижний предел на массу таких частиц равен 1590 ГэВ.

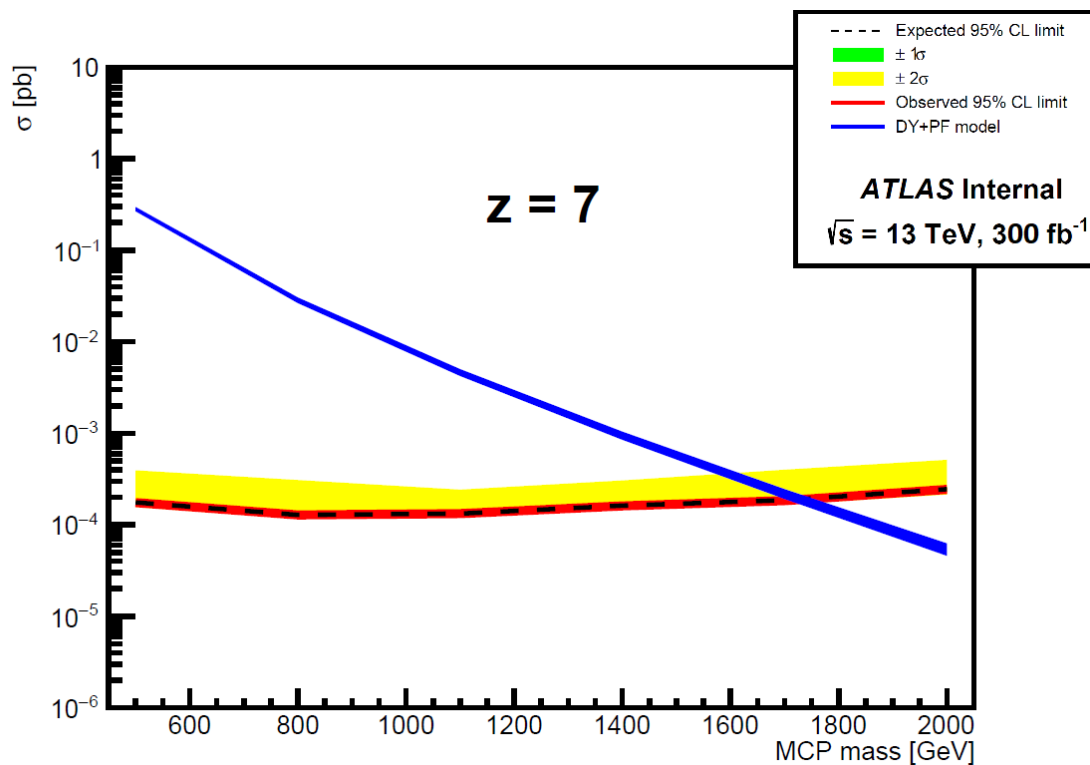


Рисунок 6.2 – Верхние пределы сечения рождения многозарядных частиц в зависимости от их массы при светимости в 300 фб⁻¹

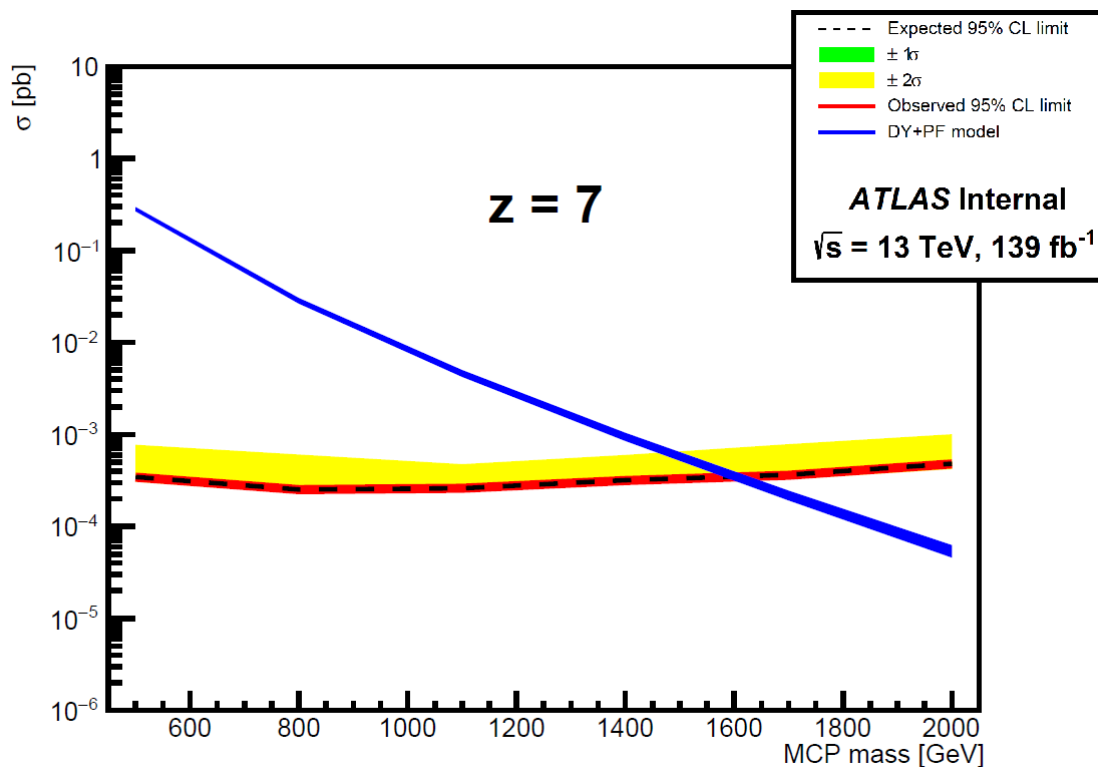


Рисунок 6.3 – Верхние пределы сечения рождения многозарядных частиц в зависимости от их массы при увеличении эффективности на 10%

При увеличении интегральной светимости со 139 фб^{-1} до 300 фб^{-1} (при прочих равных условиях) этот график примет следующий вид (см. рис. 6.2). Нижний предел на массу искомым частиц увеличится с 1590 ГэВ до 1730 ГэВ .

Наконец, предположим, что при номинальной светимости в 139 фб^{-1} эффективность увеличится на 10% (при прочих равных условиях). Реально увеличить эффективность на 10% не представляется возможным. Но всё равно можно провести оценку влияния изменения эффективности поиска на нижний предел массы. Тогда график примет следующий вид (см. рис. 6.3). Предел на массу становится равным 1610 ГэВ .

Отдельно отметим, что все три числа (1590 ГэВ , 1730 ГэВ и 1610 ГэВ) были округлены до десяти ГэВ .

Также ещё раз заметим, что в обоих случаях меняется ровно один параметр, а все остальные остаются неизменными. Перечислим все параметры, необходимые для вычисления пределов на сечение процессов рождения многозарядных частиц с зарядом $\pm 7e$:

- интегральная светимость 139 фб^{-1}
- количество ожидаемых от фона СМ событий с хотя бы одной сигналоподобной частицей в эксп. данных:
 $(8.2 \pm 0.6(\text{стат.}) \pm 3(\text{сист.})) \cdot 10^{-2}$ событий
- предполагаемое количество наблюдаемых событий с хотя бы одной сигналоподобной частицей в эксп. данных: 0
- эффективность поиска таких частиц указана в таблице 6.1.

При изменении эффективности сохранялась её относительная погрешность, а абсолютная, соответственно, изменялась. Это справедливо и для случая изменения светимости.

Масса, ГэВ	Эффективность, %
500	5.8 ± 0.2
800	8.0 ± 0.2
1100	7.7 ± 0.2
1400	6.6 ± 0.2
1700	5.7 ± 0.2
2000	4.5 ± 0.2

Таблица 6.1 – Таблица зависимости эффективности поиска частиц с зарядом $7e$ от их массы

7 Заключение

В данной работе был проведён небольшой обзор на проблемы Стандартной модели и современной физики частиц, и в частности рассмотрен вопрос существования новых частиц, сходных по свойствам с лептонами, но имеющих электрический заряд больше $1e$ (по модулю). Также была рассмотрена работа нашей группы в эксперименте ATLAS по поиску таких частиц.

В рамках научной работы была разработана программа на языке Python для обработки результатов работы 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.

Была подготовлена возможность реинтерпретации анализа с помощью фреймворка RECAST. Т.е. существующие две части анализа и одна, разработанная в этой работе, были сохранены в образы Docker, к которым были написаны инструкции на языке yadage. Эти инструкции реализованы в двух разных оболочках — recast-atlas и ReAna-client. Отдельно образы и весь анализ в целом были протестированы для частных случаев, в частности на ОС Windows.

Проведя весь цикл анализа для вычисления верхнего предела на сечение процесса рождения многозарядных частиц массой 500 ГэВ и элементарным зарядом $\pm 7e$, на малой статистике получен наблюдаемый верхний

предел на сечение рождения (0.41 фб), близкий по значению к пределу, полученному в анализе, выполненном традиционными средствами (0.39 фб), что подтверждает правильность работы ПО.

Проведена реинтерпретация анализа с целью изучить влияние изменений в эксперименте ATLAS и LHC Run-3 в целом на нижний предел массы многозарядных частиц. Полученные значения нижних пределов массы:

Исходное значение: 1590 ГэВ;

При изменении светимости со 139 фб^{-1} до 300 фб^{-1} : 1730 ГэВ;

При изменении эффективности поиска на 10%: 1610 ГэВ.

Отдельно стоит отметить, что были получены навыки сохранения и повторного воспроизведения анализа, программирования на языке Python с использованием различных библиотек, а также опыт работы с GitLab CI/CD, Docker и фреймворком RECAST (yadage, recast-atlas и ReAna-client).

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. *Khlopov M. Y., Stephan C. A.* Composite dark matter with invisible light from almost commutative geometry. — arXiv:astro-ph/0603187, 2006.
2. *Sannino F., Tuominen K.* Orientifold theory dynamics and symmetry breaking. — Vol. D71. — P. 051901 : Phys.Rev., 2005.
3. *Mohapatra R. N., Pati J. C.* Left-Right Gauge Symmetry and an Isoconjugate Model of CP Violation. — Vol. 11 : Phys.Rev.D., 1975.
4. *Jäger, et. al.* Bounds and prospects for stable multiply charged particles at the LHC // Journal of High Energy Physics. — 2019, Apr — v.2019, №4.
5. *Kerr J., Song W. Y., Taylor W.* Pair production of magnetic monopoles and stable high-electric charge objects // tech. rep. ATL-COM-PHYS-2019-1073. — 2019.
6. *Смирнов Ю. С.* Результаты поиска долгоживущих многозарядных частиц в эксперименте ATLAS на БАК ЦЕРН. — Диссертация на соискание учёной степени кандидата физико-математических наук, 2017.
7. *Aaboud M., et. al.* Search for heavy long-lived multi-charged particles in pp collisions at $\sqrt{s} = 13 \text{ TeV}$ using the ATLAS detector // Physical Review D. — 2019, Mar — v.99, №5.
8. *Smirnov Y., et. al.* Search for heavy long-lived multi-charged particles in pp collisions in full Run-II data at $\sqrt{s} = 13 \text{ TeV}$ using the ATLAS detector // tech. rep. ATL-COM-PHYS-2020-141. — 2021.

9. *Kyle Cranmer, et. al.* Analysis Preservation in ATLAS. — J. Phys.: Conf. Ser.664 032013, 2015.
10. *Solomon Hykes.* Docker. — <https://www.docker.com/>, 2013.
11. *Lukas Heinrich.* yadage – yaml based adage. — <https://yadage.readthedocs.io/>, 2016.
12. *HEP Software Foundation.* ReAna. — <https://reanahub.io/>, 2019.
13. *Richard Hipp D.* SQLite3. — www.sqlite.org/index.html, 2000.
14. *Vorobyev G. I.* Uncertainty Summation. — gitlab.cern.ch/atlas-phys/exot/ueh/MultiChargedParticles/2015-2018/UncertaintySummation, 2015.
15. *Smirnov Y. S.* Search for multi-charged particles in the 2015-2018 data. — gitlab.cern.ch/atlas-phys/exot/ueh/MultiChargedParticles/2015-2018, 2021.
16. *Smirnov Y. S., Vorobyev G. I., Macdonell D. M.* ANA-EXOT-2018-54. — <https://gitlab.cern.ch/recast-atlas/exotics/ana-exot-2018-54>, 2021.
17. *ATLAS Collaboration.* Technical Design Report New Small Wheel : tech. rep. / CERN - LHCC. — 2013.
18. *ATLAS Collaboration.* The New Small Wheel Upgrade Project of the ATLAS Experiment : tech. rep. / Simon Fraser University, University Dr. Burnaby. — 2014.