

МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ
УЧРЕЖДЕНИЕ ОБРАЗОВАНИЯ
«Гомельский государственный технический университет имени П.О.
Сухого»

КАФЕДРА «ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ»

РЕФЕРАТ

на тему

**ПРОГРАММНЫЙ КОМПЛЕКС ДЛЯ ИМИТАЦИОННОГО
МОДЕЛИРОВАНИЯ РОЖДЕНИЯ Z' - БОЗОНОВ В
ПРОТОН-ПРОТОННЫХ СТОЛКНОВЕНИЯХ С УЧЕТОМ
ЭФФЕКТОВ $Z - Z'$ СМЕШИВАНИЯ**

подготовленный для прохождения итоговой аттестации по
общеобразовательной дисциплине «Основы информационных технологи»

Выполнил:

магистрант гр. МАГ 40-22 специальности 1–40 80 04 «Математическое
моделирование, численные методы и комплексы программ»

Бурим Илья Павлович

Проверил:

доцент кафедры «Информационные технологии»

Цитринов А.В.

Гомель 2017

СОДЕРЖАНИЕ

Перечень условных обозначений и сокращений	3
ВВЕДЕНИЕ	4
ОБЩАЯ ХАРАКТЕРИСТИКА РАБОТЫ	5
1 Анализ программ моделирования процессов столкновения элементарных частиц при высоких энергиях	8
1.1 Инструменты имитационного моделирования	8
1.2 Обзор генератора «PYTHIA»	10
1.3 Обзор генератора «Powheg-Box»	12
1.4 Обзор генератора «Sherpa»	16
2 РАЗРАБОТКА АЛГОРИТМОВ И ТЕХНОЛОГИЙ РЕШЕНИЯ ПОСТАВЛЕННОЙ ЗАДАЧИ	19
2.1 Рождения Z' - бозонов в протон-протонных столкновениях с учетом эффектов Z - Z' смешивания	19
2.2 Используемые средства разработки программного обеспечения	30
3 РАЗРАБОТКА <i>WEB</i> -ПРИЛОЖЕНИЯ	39
4 ТЕСТИРОВАНИЕ И ВЕРИФИКАЦИЯ РАЗРАБОТАННОГО ПРИЛОЖЕНИЯ	49
4.1 Верификация работы программы	49
4.2 Анализ результатов верификации	54
ЗАКЛЮЧЕНИЕ	57
БИБЛИОГРАФИЧЕСКИЙ СПИСОК	58
ПРИЛОЖЕНИЕ А	60

Перечень условных обозначений и сокращений

В настоящей пояснительной записке применяются следующие термины, обозначения и сокращения.

БАК – Большой Адронный Коллайдер.

ДЯ – Дрелл-Янга.

КХД – Квантовая хромодинамика, калибровочная теория сильных взаимодействий.

ЛЭП – большой электрон-позитронного коллайдер.

НФ – Новая Физика.

СМ – Стандартная Модель.

СУБД – Система управления базами данных

ATLAS (A Toroidal LHC ApparatuS) – один из четырёх основных экспериментов на Большом адронном коллайдере в Европейской организации ядерных исследований CERN в городе Женева (Швейцария).

AWS (Amazon Web Services) – коммерческое публичное облако, поддерживаемое и развиваемое компанией Amazon.

CMS (Compact Muon Solenoid) – один из двух больших универсальных детекторов элементарных частиц на Большом адронном коллайдере.

HCI (Human-Computer Interaction) – взаимодействие человека и компьютера.

JVM (Java Virtual Machine) – виртуальная машина Java — основная часть исполняющей системы Java.

JIT (Just-In-Time compiler) – часть виртуальной машины Java, которая используется для ускорения выполнения приложений.

JMX (Java Management Extensions) – технология Java, предназначенная для контроля и управления приложениями.

LEP (Large Electron-Positron collider) – ускоритель заряженных частиц осуществляющий столкновения электронов и их античастиц – позитронов.

LHC (Large Hadron Collider) – ускоритель заряженных частиц на встречных пучках, предназначенный для разгона протонов и тяжёлых ионов и изучения продуктов их соударений.

MVC (Model-View-Controller) – схема разделения данных приложения, пользовательского интерфейса и управляющей логики на три отдельных компонента.

NLO (Next-to-leading order) – возмущающие вычисления КХД следующего порядка.

SLC (Stanford Linear Collider) – линейный коллайдер сталкивающий электроны и позитроны каждый с энергией до 50 ГэВ.

SMC (Shower Monte Carlo) – метод моделирования событий в физике высоких энергий при помощи генератора случайных величин.

SSM (Supersymmetric Standard Model) – идея суперсимметрии в стандартной модели – симметрии между фермионами и бозонами.

SVG (Scalable Vector Graphics) – язык разметки масштабируемой векторной графики предназначен для описания двумерной векторной и смешанной векторно или растровой графики в формате XML.

UI (User Interface) – коммуникационный канал, по которому осуществляется взаимодействие пользователя и компьютера.

ВВЕДЕНИЕ

Одной из основных задач современной теоретической и экспериментальной физики является проверка Стандартной модели электрослабых и сильных взаимодействий элементарных частиц (СМ) [1-15], которая осуществлялась в ускорительных экспериментах на высокоэнергетических коллайдерах, таких как *LEP*, *SLC*, *Tevatron*, *HERA* и др., а также интенсивно ведется в настоящее время на Большом адронном коллайдере *LHC*. Последний громкий успех СМ связан с открытием хиггсовского бозона в экспериментах *CMS* и *ATLAS* на *LHC*. Для более детального исследования свойств хиггсовского бозона планируются новые коллайдерные эксперименты, такие как проекты *ILC* и *CLIC*. Стандартная модель не объясняет, что такое гравитация и как она связана с другими силами и частицами. Также она не объясняет, почему основными частицами вещества являются кварки и лептоны и сколько их должно быть. Кроме этого Стандартная модель не объясняет таких явлений, которые по праву должны учитываться при больших энергиях, а теперь исследуются ускорителями частиц. Одно из таких явлений – «темная материя». По последним данным считается, что доминирующей формой материи во Вселенной является так называемая «Темная материя». Без темной материи галактики и звезды не сформировались бы и жизни не существовало бы. Только в последние 10-15 лет ученые добились существенного прогресса в понимании свойств темной материи. Недавние наблюдения влияния темной материи на структуру Вселенной показали, что она отличается от любой формы материи, которую обнаружили или измерили в лаборатории. В то же время появились новые теории, которые могут сказать нам, что такое темная материя. В настоящее время на современных ускорителях элементарных частиц ведутся поиски кандидатов на частицы темной материи. Если эти частицы имеют массы, которые измеряются в шкале ТэВ, то они могут быть обнаружены на Большом адронном коллайдере. Однако проверка того, что эти новые частицы действительно связаны с темной материей, потребует, получение их характеристик.

ОБЩАЯ ХАРАКТЕРИСТИКА РАБОТЫ

Связь работы с научными программами (проектами) и темами

Диссертационная работа связана с тематикой НИР, выполняемых в рамках научно-исследовательского направления кафедры «Информационные технологии» Гомельского государственного технического университета им. П. О. Сухого. Тема диссертации соответствует приоритетным направлениям фундаментальных исследований в Республики Беларусь. Диссертационная работа выполнялась в период с 2017 по 2019 годы в рамках отдельного подзадания государственной программы научных исследований «Конвергенция-2020», подзадание 2.1.05, номер гос. регистрации 20162284.

Цель и задачи исследования

Целью работы является создание системы определяющей возможность рождения нового резонанса нейтрального спина 1 (Z') из доступных данных групп *ATLAS* для W^+W^- распадов. В качестве результатов работы будут получены ограничения на соответствующие Z - Z' -коэффициенты смешивания и на массу $M_{Z'}$. Для достижения поставленной цели были поставлены следующие задачи:

- Изучить процесс W^+W^- распадов и рождения Z' бозонов на БАК;
- Описание и создание математической модели изученного процесса;
- Создание программного обеспечения для вычисления ограничений на соответствующие Z - Z' -коэффициенты смешивания и на массу $M_{Z'}$;
- Построение *web*-приложения для демонстрации результатов;

Изучение появления электрослабых бозонов дает мощную проверку спонтанного нарушения калибровочной симметрии стандартной модели и может быть использовано для поиска новых явлений за пределами стандартной модели. Дополнительные нейтральные векторные бозоны Z' , распадающиеся на заряженные пары калибровочных векторных бозонов W^+W^- , прогнозируются во многих сценариях новой физики, включая модели с расширенным калибровочным сектором.

Научная новизна

Научная новизна работы заключается в том, что впервые получены ограничения на угол смешивания Z' -бозонов в процессе рождения W^+W^- пар в протон-протонных столкновениях для светимостей 1000 фб^{-1} и 3000 фб^{-1} на Большом Адронном коллайдере, а также создан программный модуль, позволяющий выполнять: имитационное моделирование рождения Z' в процессе W^+W^- на Большом Адронном коллайдере с учётом эффектов Z - Z' смешивания.

Положения, выносимые на защиту

Автором защищаются:

- имитационная модель процесса рождения Z' -бозонов в протон-протонных столкновениях с учетом эффектов Z - Z' смешивания;
- программный модуль для имитационного моделирования процесса рождения Z' -бозонов в протон-протонных столкновениях с учетом эффектов Z - Z' смешивания в условиях выполненных экспериментов на *ATLAS*;
- оценки ограничений на углы смешивания Z' -бозонов в процессе рождения W^+W^- пар в протон-протонных столкновениях в условиях экспериментов на Большом адронном коллайдере, рассчитанные для интегральной светимости 1000 фб^{-1} и 3000 фб^{-1} , которые составили 10^{-4} и 6×10^{-5} , соответственно, и существенно превышают существующие экспериментальные ограничения.

Личный вклад соискателя

Научные и практические результаты диссертации, положения, выносимые на защиту, разработаны и получены лично соискателем или при его непосредственном участии.

Апробация результатов диссертации

Результаты работы докладывались на студенческой международной научно-практической конференции «Научное сообщество студентов XXI столетия. ТЕХНИЧЕСКИЕ НАУКИ» номер LXXIX.

Опубликованность результатов диссертации

Результаты диссертационных исследований, связанных с измерением процесса рождения W^+W^- пар в протон-протонных столкновениях и получены экспериментальные ограничения на угол смешивания Z' -бозонов ожидают публикаций.

Структура и объем диссертации

Диссертационная работа состоит из введения, четырёх глав, заключения и библиографического списка. Объем диссертации – 75 листов, включая 3 приложения и 46 иллюстраций. Библиографический список содержит 18 наименований, так же 1 публикацию соискателя.

ГЛАВА 1

Анализ программ моделирования процессов столкновения элементарных частиц при высоких энергиях

1.1 Инструменты имитационного моделирования

Физика высоких энергий — передовое направление современной науки, конечной целью которого является открытие наиболее фундаментальных законов микромира, управляющих эволюцией материи во Вселенной, начиная с момента ее рождения при Большом взрыве. Физика высоких энергий встречает XXI век реализацией гигантского проекта Большого адронного коллайдера (БАК) [8]. Этот уникальный, не имеющий себе равных по масштабам и сложности, научный проект, который находится сейчас в процессе реализации международным сообществом физиков из более чем 40 стран на базе европейской организации ядерных исследований, базирующейся в Женеве, направлен на решение краеугольных проблем современной субъядерной физики.

Для исследования отклика детектора на различные физические процессы, созданы программы, позволяющие перевести моделированное на уровне частиц событие взаимодействия протонов при соударении в формат представления данных детекторов установки *ATLAS*. Алгоритмы моделирования интегрированы в программную оболочку эксперимента *ATLAS*, именуемую *Athena*, использующую программный пакет *GEANT4*.

Генератор события создает набор частиц, который направляется в программу быстрого или полного моделирования детектора. Генераторы событий встроены в *Athena*. Используется большое число других, поддерживаемых авторами, генераторов, которые имеют блоки связи для использования в *Athena*. Основной массив модельных событий создан с помощью генераторов *PYTHIA* [11], включая его версию *PYTHIAB*, предназначенную в *ATLAS* для моделирования событий с рождением *B*-адронов.

PYTHIA - это программного пакета для визуализации результатов моделирования процессов столкновения частиц при высоких энергиях осуществляющего генерацию методом Монте-Карло физических событий.

Программы *PYTHIA* интенсивно используются для генерации событий в физике высоких энергий при описании процессов множественного рождения в столкновениях элементарных частиц. В частности задачи, что включает решаемые жесткие с взаимодействия помощью данного в столкновении

ях e^+e^- , pp и ep , а также некоторые другие случаи. Программа предназначена для генерации генератора полных событий, т.е. дают более детальную картину, чем мы наблюдаем в эксперименте, в рамках нашего понимания фундаментальной физики процессов. Обсуждаемые здесь программы Монте-Карло построены как ведомые системы, т.е. пользователь должен написать основную программу. Из нее различные программы вызываются для выполнения частных задач, после чего управление снова передается основной программе. Некоторые из этих задач могут быть весьма тривиальными, и достаточно высокоуровневые программы могут производить большое число вызовов подпрограмм.

Генераторы общего назначения создают событие как целое. Они используют много параметров, часть из которых относится к фундаментальным параметрам, такие как константы связи квантовой хромодинамики (КХД) и электрослабой теории, часть относится к моделям, описывающим взаимодействия на больших расстояниях, с малыми передачами импульса, т.н. «мягкой» КХД, и к электрослабым процессам.

Для корректного моделирования процессов рождения и распада частиц необходимо учитывать условия проведения эксперимента. Это условия рождения изучаемых частиц на ускорителе при соответствующих энергиях сталкивающихся пучков, полные цепочки распадов частиц до уровня «стабильных частиц», регистрируемых детектором. Для решения этих задач применяются генераторы событий, использующие метод Монте-Карло. Генератор *PYTHIA* является широко используемой в физике высоких энергий программой моделирования столкновений различных частиц в широком диапазоне энергий. Этот генератор учитывает процессы фрагментации кварков в адроны и разыгрывает сложные цепочки адронных распадов. Стартуя с заданного пользователем процесса, (столкновение двух протонов с рождением Z -бозона и т.п.) программа случайным образом (с учетом законов сохранения и, по возможности, теоретически известной структуры взаимодействия) разыгрывает конфигурацию конечных партонов, а затем моделирует т.н. процесс адронизации - процесс превращения ненаблюдаемых кварков и глюонов в реальные стабильные и нестабильные частицы с последующим распадом нестабильных частиц. На выходе программа выдает список всех частиц, родившихся в результате столкновения заданных первичных частиц, значения их компонент импульса и энергии. Кроме того, имеется возможность проследить последовательность рождений и распадов от первичного взаимодействия до рождения данной частицы. В качестве входных параметров программы используются описания сталкивающихся частиц, их энергий и тип моделируемого процесса (например, рождение Z -бозона). Существующие версии

пакета *PYTHIA* написаны на языке программирования *FORTRAN*. Результаты генерации – характеристики вторичных частиц – записываются в файл, что позволяет в дальнейшем проводить статистическую обработку событий.

1.2 Обзор генератора «PYTHIA»

PYTHIA это программа для генерации событий физики высоких энергий, т.е., для описания столкновений таких высокоэнергетических элементарных частиц, как электрон, позитрон, протон и антипротон в различных комбинациях. Информация для моделирования взята по большей части из собственных исследований в ЦЕРН, однако, много формул и другой информации почерпнуто из научной литературы [1].

С 1997 года по настоящее время использовалась версия этого Монте-Карло генератора, написанная в *FORTRAN77* (текущая версия 6.4). Сейчас программа переписана в *C++* (версия 8.1), однако, до тех пор, пока не осуществлен перевод всех возможностей, обе версии используются и поддерживаются одновременно.

Назначение генераторов физических событий:

- Дают физикам представление о типе событий, которые они надеются увидеть, и об их скорости набора;
- Помогают в планировании новых детекторных установок, то есть, оптимизировать их характеристики для изучения интересующих сценариев физических событий в рамках существующих ограничений;
- Являются инструментом для проработки стратегии анализа данных (оптимизации отношения “сигнал/шум”);
- Используются в качестве метода оценки коррекций на геометрические и кинематические ограничения области чувствительности (acceptance) детекторов;
- Используются в качестве удобной рабочей оболочки для интерпретации наблюдаемых феноменов в терминах Стандартной Модели.

Квантовая механика вносит концепцию случайности в поведение физических процессов. Достоинством генераторов событий (*event generators*) является то, что эта случайность может быть смоделирована при помощи метода Монте-Карло. Сущность метода заключается в том, что, во первых подразумевается наличие генератора псевдослучайных чисел, т.е. функции, которая при вызове возвращает число R в пределах от 0 до 1, при этом

распределение R является плоским, и с достаточной точностью значения R являются нескоррелированными [1]. Затем эти значения R используются для розыгрыша сценария конкретного цикла события (выбор конкретного значения для различных известных распределений величин, выбор времени распада и т.п.) Разыграв статистически достаточное количество событий, мы можем построить интересующие нас распределения (например, диапазон энергий для продуктов интересующего механизма реакции). Что касается упомянутых известных распределений, то, например, дифференциальное сечение реакции рассчитывается из кинематических соотношений при введении матричного элемента (известного, либо предложенного теоретиками исходя из перспективных моделей), для учета высших порядков КХД вводится значение дополнительного параметра (К-множителя). Следует заметить, что при генерации значительного числа событий (миллионов), становится актуальной проблема скоррелированности псевдослучайных чисел, и вместо встроенных в *C++ Random* генераторов приходится применять специально созданные программы.

С точки зрения описания физики событий полная процедура генерации события разделяется на 3 стадии:

1. Генерация «процесса», который определяет природу события. Зачастую это могут быть «жесткие процессы», такие как $gg \rightarrow h^0 \rightarrow ZZ \rightarrow t^+t^-qq_{bar}$ (а также другие процессы), которые могут быть просчитаны в рамках теории малых возмущений.
2. Генерация всех подчиненных процессов на партонном уровне, включая гамма-излучение, многократные партонные взаимодействия и структуру непервзаимодействовавшего пучка. Такие феномены приблизительно описываются теорией малых возмущений, однако непертурбативные поправки уже существенны.
3. Адронизация этой партонной конфигурации (фрагментация струй, распады нестабильных частиц). Только феноменологическое описание

Этим стадиям отвечают три класса *ProcessLevel*, *PartonLevel* и *HaronLevel*, соответственно. Классы: *Event* (члены класса *process* и *event*), *BeamParticles*, база данных *Settings*

С точки зрения технического устройства, взаимодействия пользователя и генератора проявляется в трех фазах:

1. Инициализация, когда формулируется задание.
2. Генерация индивидуального события (цикл события).

3. Вывод окончательной статистики.

Программа содержит теорию и модели для ряда аспектов физики, включая так называемые мягкие и жесткие взаимодействия, распределения партонов, партонные струи начального и конечного состояний, многократные партонные взаимодействия, фрагментацию и распады [1].

Встроенные *C++* методы программы обеспечивают доступ к информации как об отдельной частице либо процессе на любом этапе розыгрыша события, так и о событии в целом. Встроенные средства вывода позволяют получить статистическую информацию и гистограммы в виде *ASCII* кода (который можно сохранить в файл для дальнейшего использования).

Пакет *Pythia* является компактной (5 Мб) независимой программой, поэтому несложно скачать и установить себе собственную локальную версию. Для этого архив установочный код (в 2011 году был *8145.tgz*) можно скачать с сайта разработчика или с кафедрального сайта [1]. Затем его нужно распаковать (команда *tar -xzf 8145.tgz*) и скомпилировать (команда *make* в распакованной корневой директории, занимает 2 минут на *lx*).

Быстрее всего начать работу с обучающими и своими первыми скриптами в поддиректории *examples*. В пакеты *Pythia* можно создать такой объемный процесс, как генератор моделирующий столкновение барионов на *LHC* с рождением топ-кварков. Так же в на сайте разработчиков библиотеки есть руководство в котором описываются возможности по настройке свойств процессов и получения информации (например, можно сделать бозон Z^0 стабильной частицей инструкцией *pythia.readString("23:onMode=off")*).

1.3 Обзор генератора «Powheg-Box»

Возмущающие вычисления КХД следующего порядка *Next-to-leading order* (*NLO*), а также *Shower Monte Carlo* (*SMC*) программы являются фундаментальными инструментами современной феноменологии физики элементарных частиц. В частности, программы *SMC* включают описание общего адронного столкновения высокой энергии процесс, начиная от столкновения между составляющими и развития партонного потока, что увеличивает количество частиц в конечном состоянии за счет сильно упорядоченных последующих выбросов [2]. В конце концов, интерфейс с феноменологической моделью адронизации позволяет сравнивать с экспериментальными данными. По этим причинам они обычно используются экспериментаторами для моделирования сигнальных и фоновых процессов в физических поисках. Тем не менее, спрос на лучшие и лучшие прогнозы для экспериментов с высокой энергией требуют повышения точности существующих *SMC*, включая исправления *NLO*. Метод *MC@NLO* [2]

сначала показал, как достичь точности NLO для инклюзивных количеств, реализуя жесткий подпроцесс в NLO и развитие ливней в ведущем логарифмическом приближении, избегая двойного счета излучений. Таким образом можно достичь преимуществ обоих подходов: генерация исключительных конечных состояний SMC и точность расчетов NLO .

Метод $POWHEG$ - это другое предписание для сопряжения вычислений NLO с партоном душевые генераторы. Этот метод не зависит от программы Монте-Карло, используемой для последующего принятия потока частиц и генерирует только положительные взвешенные события. В этом отношении это улучшает подход $MC@NLO$. До сих пор метод $POWHEG$ был успешно применен к нескольким процессам, как на лептонные и адронные коллайдеры [2]. В этих реализациях это подключен к программам $SMC HERWIG$, $PYTHIA$ и $HERWIG ++$.

В методе $POWHEG$ самое сильное излучение генерируется первым, независимо от следующие. Схематически самое жесткое излучение распределяется в соответствии с

$$d\sigma = \bar{B}(\Phi_B)d\Phi_B[\Delta_R(p_T^{min}) + \frac{R(\Phi_R)}{B(\Phi_B)}\Delta_R(k_T(\Phi_R))d\Phi_{rad}],$$

где $B(\Phi_B)$ - вклад Борна и

$$\bar{B}(\Phi_B) = B(\Phi_B) + [V(\Phi_B) + \int d\Phi_{rad}R(\Phi_R)]$$

является дифференциальным сечением NLO при фиксированной основной кинематике Борна и интегрированной по радиационные переменные. Поперечный импульс испускаемого партона относительно пучка или другой частицы, в зависимости от особенности области, обозначается через $k_T(\Phi_R)$. нижний предел p_T^{min} необходимо, чтобы константа связи не достигала нефизических значений. $V(\Phi_B)$ и $R(\Phi_R)$ являются виртуальными и действительными поправками и в выражении внутри квадратной скобки в формуле (2) процедура, которая заботится об отмене мягких и коллинеарных особенностей, например, *Fraxione-Kunszt-Signer (FKS)* или Катани-Сеймур (*CS*) дипольное вычитание. затем,

$$\Delta_R(P_T) = exp[- \int d\Phi_{rad} \frac{R(\Phi_R)}{B(\Phi_B)} \theta(k_T(\Phi_R) - p_T)]$$

это $POWHEG$ Судаков, то есть вероятность того, что выброс не будет тяжелее, чем p_T . Уравнение (1) можно рассматривать как улучшение исходной формулы для наиболее сложных выбросов SMC , поскольку сечение Борна заменяется на $\bar{B}(\Phi_B)$, которое по построению нормированы на NLO . При малых поперечных импульсах $POWHEG CS$ становится равным

стандартному *SMC*. Тем не менее, *NLO* область излучения с высоким p_T правильно описывается реальными вкладом:

$$d\sigma \approx \bar{B}(\Phi_B) d\Phi_B \frac{R(\Phi_R)}{B(\Phi_B)} d\Phi_{rad} \approx R(\Phi_R) d\Phi_B d\Phi_{rad}$$

поскольку $\Delta_R \approx 1$ и $\bar{B}/B \approx 1 + \Theta(\alpha_s)$. После генерации самого жесткого излучения можно интерфейс с любым доступным генератором столкновений, для того, чтобы обработать остальной поток, чтобы избежать двойной регистрации частиц, *SMC* должен быть либо p_T -упорядоченным, либо иметь возможность наложить вето на выбросы с p_T сложнее, чем первый.

В реальном процессе столкновения присутствуют несколько цветных безмассовых партонов, либо в начальном, либо в конечном состоянии. Таким образом, следует повторить процедуру, изложенную в начале главы для каждого возможного единственного числа областей, связанных с любой безмассовой цветной веткой, становящейся коллинеарной к другой, или мягкой. Для этого всё реальное сечение эмиссии раскладывается в сумму слагаемых, каждое из которых имеет не более одной коллинеарной и одной мягкой особенности. Затем излучение генерируется независимо в каждом из этих регионов, но сохраняется только самое сильное излучение, и событие генерируется в соответствии с ароматом и кинематикой, связанной с ним. Из-за этой сложности, автоматический инструмент, *POWHEG-BOX*, был построен [2], чтобы помочь включению новых процессов. С другой стороны, *POWHEG-BOX* также может рассматриваться как библиотека, где ранее реализованные процессы доступны в общей структуре. Процессы реализованы так далеко и уже доступны в публичной версии включают: W , Z/γ производство одного вектора бозона, Бозон Хиггса через глюон и вектор бозон-фьюжн, однолучевой в s - и t -каналы.

Пользователь, желающий включить новый расчет *NLO*, должен знать только, как сообщить нужную информация для *POWHEG-BOX*. Это происходит либо путем определения соответствующих переменных, либо предоставляя необходимые процедуры Фортрана. Требуемые входы:

1. Количество ветвей в процессе Борна, например, $n_{legborn} = 5$ для $pp \rightarrow (Z \rightarrow e^+e^-)j$
2. Список ароматов *Born* и *Real*, согласно соглашениям PDG [2], аромат определен входящий (исходящий) для входящих (исходящих) фермионных линий, например, для $bu \rightarrow Ztsg$.
3. Процедура Борновского фазового пространства, которая, учитывая случайные числа в единицах измерения $ndims$ гиперкуба, задает

борновское фазовое пространство якобиана и возвращает импульсы в неизвестных x .

4. Подпрограммы, выполняющие инициализацию соединений и настройку шкалы факторизации и перенормировки.
5. Процедура амплитуды Борна в квадрате, для заданного набора импульсов и ароматов конфигурации, возвращает $B = |M|^2$ суммируется и усредняется по цвету и спирали как упорядоченные по цвету квадраты Борна, амплитуды B_{jk} и спиральность коррелировали по квадрату Борна амплитуды $B_{k,\mu\nu}$, где k пробегает все внешние глюоны.
6. Подпрограмма квадрата амплитуды реального излучения, которая возвращает R для заданных импульсов и список ароматов.
7. Конечная часть интерференции борновского и виртуального амплитудных вкладов $\nu_b = 2\text{Re}\{B \times V\}$ после вычета общего множителя $N = \frac{(4p)^\xi}{G(1-\xi)}(\frac{\mu_R^2}{Q^2})^\xi$. Эта рутина определяется импульсами и списком ароматов в качестве входных данных.
8. Цветовые структуры Борна в большом пределе N_c задаются через интерфейс *Les Houches* [2].

Пункты (1-7) являются обычными ингредиентами, необходимыми для выполнения расчета NLO в любом методе вычитания. Элемент (8) вместо этого необходим для обеспечения определенной цветовой структуры генератор SMC . Внутри *POWHEG-BOX* реализована процедура вычитания FKS . В начале пакет автоматически оценивает комбинаторику, выявляя все особые области и соответствующие базовые вклады Борна. Он также выполняет проекцию реальных вкладов на особую область и вычисляет вычитание контртермы из мягких и коллинеарных приближений реальных выбросов. Затем пакет строит ISR и FSR фазовые пространства, согласно FKS параметризации особой области и выполняет интеграцию. В конце концов, каждый получает дифференциальное сечение NLO . На данном этапе, можно также взаимодействовать с некоторой процедурой анализа, чтобы получить дифференциальные распределения NLO как побочный продукт. После этапа интеграции выполняется вычисление верхних границ для эффективная генерация событий, подавленных эффектом Судакова, а затем генерация сильнейшего излучения. На данный момент генерируются события, которые содержат не более одного излучения, которое должны быть переданы в стандартную программу SMC , для разработки остальных.

1.4 Обзор генератора «Sherpa»

Наиболее яркими примерами генераторов событий являются очень успешные, хорошо отлаженные программы *PYTHIA* и *HERWIG*. Они были построены за последние десятилетия наряду с экспериментальными открытиями, и большинство особенностей, видимых в прошлых и настоящих экспериментах, могут быть описаны ими. Тем не менее, необходимость в более высокой точности для решения задач новых энергетических масштабов происходящих на *LHC*, сложность конечных состояний в этих масштабах, необходимость обслуживания и желание легкой возможности реализовать новые физические модели, требовалось переписать эти пакеты на современном языке программирования, обеспечивающем более высокий уровень модульности.

Объектно-ориентированные рамки отвечают последним требованиям и в связи с предпочтениями сообщества к *C++*, новое поколение генераторов событий построено на этом языке программирования [3]. Это привело к улучшению повторных реализаций в форме программ *PYTHIA 8* и *HERWIG ++* - преемники версий написанных на *Fortran* упомянутых выше и к созданию генератора событий *SHERPA* [3]. В связи с этим в последнее десятилетие стали доступны пакеты для расчетов с опережающим порядком. Яркими примерами являются *MCFM* и *NLOJET ++*. Соответствующие методы реализованы, например, в *MC@NLO*, который основан на фортранской версии *HERWIG*, в *HERWIG ++* и в некоторых более специализированных программах [3]. Тем не менее, полные расчеты следующего за ведущим порядком, лежащие в основе этих новых методов, очень сложны, и до сегодняшнего дня контролируются только процессы с пятью внешними ветвями. Но с другой стороны, многие важные экспериментальные сигнатуры зависят от конечных состояний с более высокой кратностью, что инициировало существенную деятельность по совершенствованию методов и инструментов с точностью на уровне множества ветвей, так что теперь доступно несколько пакетов, которые могут вычислять соответствующие сечения и генерировать события полностью автоматизированным способом. Наиболее яркими примерами являются *ALPGEN*, *CompHEP / CalcHEP*, *HELAC-PHEGAS*, *MADGRAPH*, *WHIZARD* и *AMEGIC ++*. В настоящее время только библиотека *AMEGIC ++* реализует данный функционал и встроена в полноценный генератор событий, а именно в среду *SHERPA*. Чтобы преобразовать многочастичные события на уровне партонов, которые предоставляются этими инструментами в ведущем порядке, в события уровня адронов, было разработано несколько алгоритмов, все из которых направлены на сохранение логарифмической точности партонного потока

и дополнения его точным результатом возмущающего ведущего порядка для учитывания кратности струи.

SHERPA [3] является аббревиатурой от «Simulation of High Energy Reactions of Particles». Программа является полная структура генерации событий, которая была построена с нуля и полностью написана в современный объектно-ориентированный язык программирования *C++*

Построение *SHERPA* осуществлялось способом, в значительной степени определяемым следующими тремя парадигмами:

- Модульность. Различные физические аспекты реализованы в почти независимых модулях, опираясь на небольшое количество структурных и вспомогательных модулей, таких как, например, запись событий и т. д. Модульность позволяет, например, иметь более одного генератора матричных элементов или параллельного потока, на выбор пользователя. Центральный модуль, *SHERPA*, управляет взаимодействием всех других частей и фактической процедуры генерации.
- Снизу-вверх. Физические модули обычно разрабатываются сами по себе, проходят испытания и проверяются, прежде чем они будут включены в полную структуру генерации событий. Это в свою очередь приводит к довольно гибкой, минимальной структуре, лежащей в основе организации генерации событий.
- Разделение интерфейса и реализации. Для того, чтобы облегчить два требования выше, *SHERPA* опирается на структуру, в которой (почти независимые) физические модули доступны только через специфичные для физики обработчики. Эти обработчики помогают *SHERPA* генерировать событие на разных этапах, каждый из которых управляется определенной реализацией обработчика фазы события, такой как Сигнальный процесс или *Jet Evolution*. Примером такого взаимодействия фазы события и физики является обработчик. Обработчик - это *Matrix Element Handler*, позволяющий генерировать события на уровне партона либо встроенные жестко запрограммированные матричные элементы или генератор матричных элементов *AMEGIC++*. Обработчик актуален для двух этапов события, генерации сигнального процесса и вследствие многоструйного слияния процедуры эволюции струй.

AMEGIC++ – это генератор матричных элементов по умолчанию используемый в *SHERPA*, основанный на диаграммах Фейнмана, которые переводятся в амплитуды спиральности [3]. *AMEGIC++* использует библиотеку интеграции фазового пространства Монте-Карло *PHASIC*. Для

оценки начального состояния (обратное рассеяние лазера, начальное излучение) и интегралов. В конечном состоянии используется адаптивный многоканальный метод. По умолчанию данный генератор идет вместе с оптимизационным пакетом *Vegas* для отдельных каналов. Кроме того, поддерживается интеграцию оптимизаторов *RAMBO* и *HAAG* [3].

Эта общая структура полностью отражает парадигму генерации событий в Монте-Карло, моделирование в четко определенные, почти независимые фазы. Соответственно, каждый обработчик фазы события абстрактным образом инкапсулирует различные аспекты генерации события для реакций с высокой энергией частиц. Затем эта абстракция заменяется реальной физикой с использованием обработчиков, которые обеспечивают сбор данных о событиях. Данный пакет при генерации может не учитывать мелкие детали базовой физики и ее реализацию в форме физического модуля [3].

Таблица 1.1 – Сравнительный анализ генераторов событий физики высоких энергий

Название	Сложность реализации модели	Время модуляции 10000 столкновений, с	Размер пакета, Мб
<i>PYTHIA</i>	Легко, так как достаточно указать коэффициенты и массы частиц	5	5
<i>Powheg-Box</i>	Сложно, так как требуется более детальная настройка начальных параметров системы	10	20
<i>Sherpa</i>	Сложно, так как необходимо устанавливать и настраивать дополнительные модули	7	25

В соответствие с таблицей сравнения 1.1 было принято решение для разработки имитационной модели использовать генератор событий физики высоких энергий *PYTHIA*. Объектно-ориентированный язык программирования, используемый в пакете *PYTHIA*, и ряд готовых модулей в пакете требующих только настройчных входных параметров способствуют более простой и быстрой реализации имитационной модели для генерации одного канала событий. В пакетах *Powheg-Box* и *Sherpa* время модуляции десяти тысячи на несколько секунд выше чем в пакете *PYTHIA*, что сложится в часы при расчете значений для всех M_Z .

ГЛАВА 2

РАЗРАБОТКА АЛГОРИТМОВ И ТЕХНОЛОГИЙ РЕШЕНИЯ ПОСТАВЛЕННОЙ ЗАДАЧИ

2.1 Рождения Z' - бозонов в протон-протонных столкновениях с учетом эффектов $Z - Z'$ смешивания

Многие сценарии Новой Физики (НФ) отличной от Стандартной Модели (СМ)[1], включая модель суперпозиций и левую-правую симметричную модель, предсказывают существование новых нейтральных и заряженных калибровочных бозонов, которые могут быть найдены на текущих или будущих коллайдерах. Поиск нового нейтрального Z' и заряженного W' калибровочных бозонов является важным аспектом экспериментально-физических программ на коллайдерах больших энергий. В этой статье мы сконцентрируемся на первом бозоне.

Предоставленные лимиты большого адронного коллайдера и виртуальные эффекты ЛЭП, через интерференцию или смешивания с Z бозонами, подразумевает что любые Z' бозоны гораздо тяжелее и менее смешиваются с Z бозонами. В зависимости от рассматриваемой теоретической модели Z массы порядка 4,5 ТэВ [2] и $Z - Z'$ углов смешивания на уровне нескольких градусов исключены [3]. Угол смешивания сильно ограничен очень высокоточными экспериментами на ЛЭП и SLC . Они включают в себя измерения из формы линии Z , из лептонных отношений ветвления, нормированных на общую адронную ширину затухания Z , а также от лептонных лево-правых асимметрий. Z' , легче чем 5 ТэВ, может быть обнаружен на БАК [3] с $\sqrt{s} = 14$ ТэВ в процессе Дрелл-Янга (ДЯ) $pp \rightarrow Z' \rightarrow l^+l^- + X$, где $l = e, \mu$.

После открытия Z' -бозона на БАК через процесс ДЯ, необходимо произвести некоторую диагностику связей и смешивания $Z-Z'$, чтобы идентифицировать основную теоретическую структуру. В настоящей работе исследуются данные $ATLAS$ [4] и CMS в канале дибозона.

$$pp \rightarrow W^+W^- + X \quad (2.1)$$

Для поиска Z -бозона, который возникает, например, в популярной модели с расширенным калибровочным сектором 2.1. Анализ основан на данных о столкновениях pp при энергии центра масс $\sqrt{s} = 13$ собранных группами $ATLAS$ [4] и CMS на БАК. В частности, данные используются для поиска

Z - Z' смешивание. На *ATLAS* события W^+W^- реконструируются через их полулептонные распады W , где один W -бозон распадается на заряженный лептон ($l = e, \mu$) и нейтрино, а другой на две струи, тогда как на *CMS* W -бозон адронически распадается на две восстановленные струи.

Процесс рождения пары W^-W^+ -бозонов (1) важен для изучения электрослабой калибровочной симметрии. Общие свойства слабых калибровочных бозонов тесно связаны с нарушением электрослабой симметрии и структуры калибровочного сектора, как и существование и структура трилинейных связей. Кроме того, канал распада дибозонов Z' исследует толщину калибровочной связи между новым и калибровочными бозонами стандартной модели. Кроме того, сила связи очень влияет на элементы распада и естественную ширину такого нового калибровочного бозона. Таким образом, детальное рассмотрение процесса (1) с высокой точностью проверяет калибровочный сектор СМ и может пролить свет на бозоны, которые могут появиться за пределами СМ. Здесь мы рассмотрим возможность наблюдения Z' -бозона в W^+W^- парного процесса на БАК, который в отличие от процесса ДЯ не является основным каналом поиска, но может помочь понять происхождение новых калибровочных бозонов.

Поиски тяжелого WW резонанса были выполнены на Теватроне исследовательскими группами *CDF* и *D0*. Группа *D0* изучала резонансное рождение дибозонов до 700 ГэВ в каналах распада $lv'l'v'$ и $lvjj$ [5]. Группа *CDF* также исследовала резонанс в WW в канале распада $evjj$, что в результате привело к обнаружения нижних лимитов масс Z' и W' -бозонов, за исключением масс превышающих 900 ГэВ, зависящих от параметра смешивания.

Исследования WW -резонансов группами *ATLAS* и *CMS* с использованием, соответственно, полулептонных и адронных событий распада в pp столкновениях при 13 ТэВ устанавливают массовые пределы 3 ТэВ для этих резонансов [4].

В дипломной работе изучается возможность рождения нового резонанса нейтрального спина 1 (Z') из доступных данных групп *ATLAS* для W^+W^- распадов. В качестве результатов работы будут получены ограничения на соответствующие $Z - Z'$ -коэффициенты смешивания и на массу $M_{Z'}$. Выполнено моделирование событий рождения Z' бозонов в процессе распада на фотонную пару и моделирование событий рождения гравитонов в процессе $lv'l'v'$. Создано *web*-приложение для демонстрации результатов вычисления.

Несмотря на впечатляющий успех в описании экспериментов, Стандартная модель не может считаться окончательной теорией элементарных частиц. У нее есть свои трудности. Физики уверены, что она должна быть частью некоторой более глубокой теории строения микромира, той частью,

которая видна в экспериментах на коллайдерах при энергиях ниже примерно 1 ТэВ. Главная задача Большого адронного коллайдера — получить хотя бы первые намеки на то, что это за более глубокая теория.

Теоретики разработали большое число кандидатов на такую теорию. Все они, естественно, включают какие-то элементы, которые отсутствуют в Стандартной модели. Часто такие теории коллективно называют «Новая физика» или «За пределами Стандартной модели». На этой странице перечислены некоторые из активно изучаемых вариантов Новой физики [8].

Суперсимметрия — это гипотетическая симметрия между фермионами и бозонами. Теории, использующие эту идею, оказываются удивительно мощными, и потому именно с суперсимметрией многие связывают надежды на открытие физики за пределами Стандартной модели. Однако до сих пор не было получено ни одного убедительного доказательства в пользу того, что суперсимметрия реализуется в нашем мире. Ее поиск является одной из главных задач Большого адронного коллайдера. Константы свя-

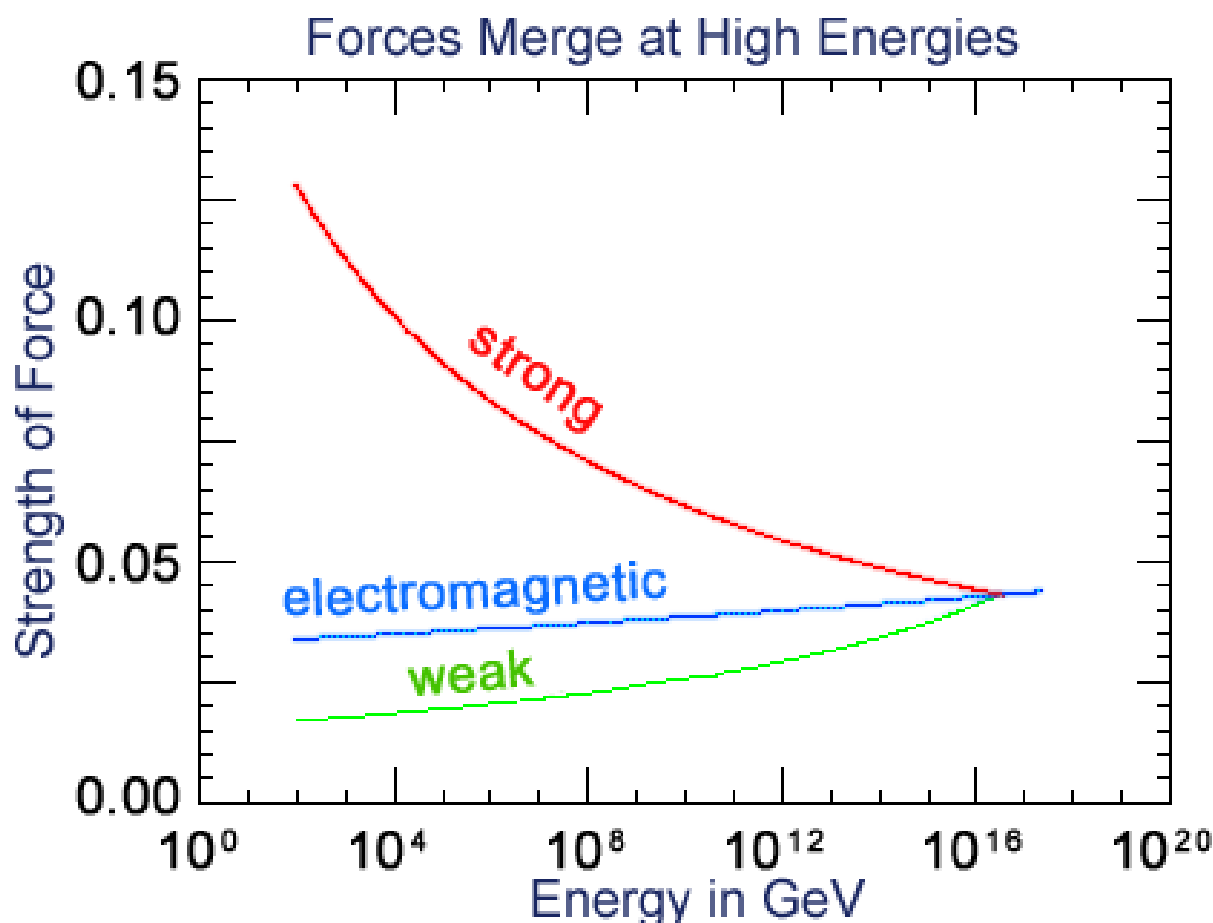


Рисунок 2.1 — Константы связи трех типов взаимодействий

зи трех взаимодействий частиц в микромире сходятся к одному значению, если имеющиеся сейчас данные экстраполировать в область очень высоких

энергий. Это совпадение считается неслучайным и воспринимается физиками как намек на то, что все три взаимодействия при больших энергиях объединяются в одно.

В XIX веке физики обнаружили, что электричество и магнетизм — это две стороны одной медали, электромагнитного взаимодействия. Век спустя, при создании Стандартной модели, электромагнетизм и слабые ядерные силы были объединены в рамках единого электрослабого взаимодействия. (Точнее говоря, внутри электрослабого взаимодействия имеются по-прежнему две разные силы, а электромагнитное и слабое взаимодействия возникают как комбинации этих сил). Каждое такое объединение упрощало теорию, уменьшало количество введенных в нее «сущностей», переводило наше понимание микромира на новый уровень.

Сейчас физики имеют сразу несколько причин подозревать, что при очень высоких энергиях происходит объединение электрослабого и сильного взаимодействий (рисунок 2.1). Модели, использующие эту идею (так называемые Теории великого объединения) разрабатываются уже давно. В идеале хотелось бы, чтобы такая теория естественным образом объясняла, почему фундаментальных взаимодействий именно столько и именно с такими свойствами, а также имела четкие предсказания, доступные проверке в современных экспериментах.

При энергиях элементарных частиц, доступных на ускорителях, гравитация по-прежнему остается исключительно слабой, так что заметить ее проявления не удастся. Однако ее сила растет с ростом энергии, и при энергиях столкновения порядка планковской она станет столь же важной, как и другие взаимодействия. В этом случае в полный рост встает исключительно сложный вопрос о том, как включить гравитацию в квантовое описание микромира. Поскольку гравитация в современной физике считается проявлением кривизны пространства-времени, успешная теория с сильной гравитацией должна описывать в рамках единого формализма не только все взаимодействия и всё вещество, но и структуру пространства-времени.

Одним из наиболее привлекательных путей решения этого вопроса является теория суперструн и ее дальнейшее развитие в виде теории бран и М-теории. В этих теориях считается, что фундаментальными объектами, существующими в многомерной вселенной, являются не точечные частицы, а протяженные объекты — струны, мембраны и еще более многомерные образования. В этой теории были получены впечатляющие успехи при высоких энергиях, однако при попытке вывести свойства нашего низкоэнергетического мира из теории суперструн возникает обескураживающая неопределенность предсказаний.

Долгое время казалось, что проверка предсказаний теории суперструн лежит далеко за пределами возможностей человечества, поскольку речь идет об энергиях, на 15 порядков превышающих энергии современных ускорителей. Однако примерно 10 лет назад возникло новое направление развития теории, в котором гравитация становится сильной на энергиях порядка 1 ТэВ. Такая возможность возникает в том случае, если наш мир более чем трехмерный и если при этом новые дополнительные пространственные размерности достаточно протяженны: либо они бесконечны, либо свернуты в многомерные петельки размером много больше ядерного масштаба.

В этом случае на *LHC* следует ожидать целый ряд совершенно замечательных эффектов, отсутствующих в Стандартной модели, например, рождение гравитонов, которые будут улетать из нашего мира в дополнительные измерения, и микроскопических черных дыр, тут же испаряющихся с испусканием множества обычных частиц. Будут также наблюдаться сильные отклонения от предсказаний Стандартной модели в столкновении обычных частиц. Стоит, впрочем, подчеркнуть, что пока нет никаких экспериментальных подтверждений того, что эта красивая гипотеза имеет отношение к нашему миру.

Все три перечисленные выше направления «Новой физики» опираются на глубокие теоретические гипотезы об устройстве нашего мира (суперсимметрия, единство сил, квантово-гравитационная вселенная). Однако кроме этих направлений теоретики также рассматривают разнообразные теории «статусом пониже». В этих теориях просто отмечается, что текущие экспериментальные данные не запрещают те или иные экзотические объекты или явления, и разрабатываются их следствия. Вот несколько примеров таких моделей разной степени экзотичности.

Неминимальные хиггсовские модели. Поскольку хиггсовские бозоны — единственные частицы Стандартной модели, до сих пор не открытые экспериментально, теоретики изучают самые разные варианты устройства этого сектора теории. Новые поколения фермионов. Можно предположить, что кроме трех известных поколений кварков и лептонов существуют и другие поколения. Частицы из этих поколений должны быть очень тяжелыми, иначе бы их уже давно открыли в эксперименте.

Новые короткодействующие силы. В таких моделях предполагается, что в нашем мире есть и иные силовые взаимодействия, отличные от сильных, слабых и электромагнитных, но они настолько короткодействующие, что до сих пор никак не проявлялись в эксперименте. На Большом адронном коллайдере благодаря его рекордной энергии удастся «прощупать» взаимодействия частиц на исключительно малых расстояниях (менее 10^{-19} метра), а значит, появляется шанс эти взаимодействия обнаружить. Они могут

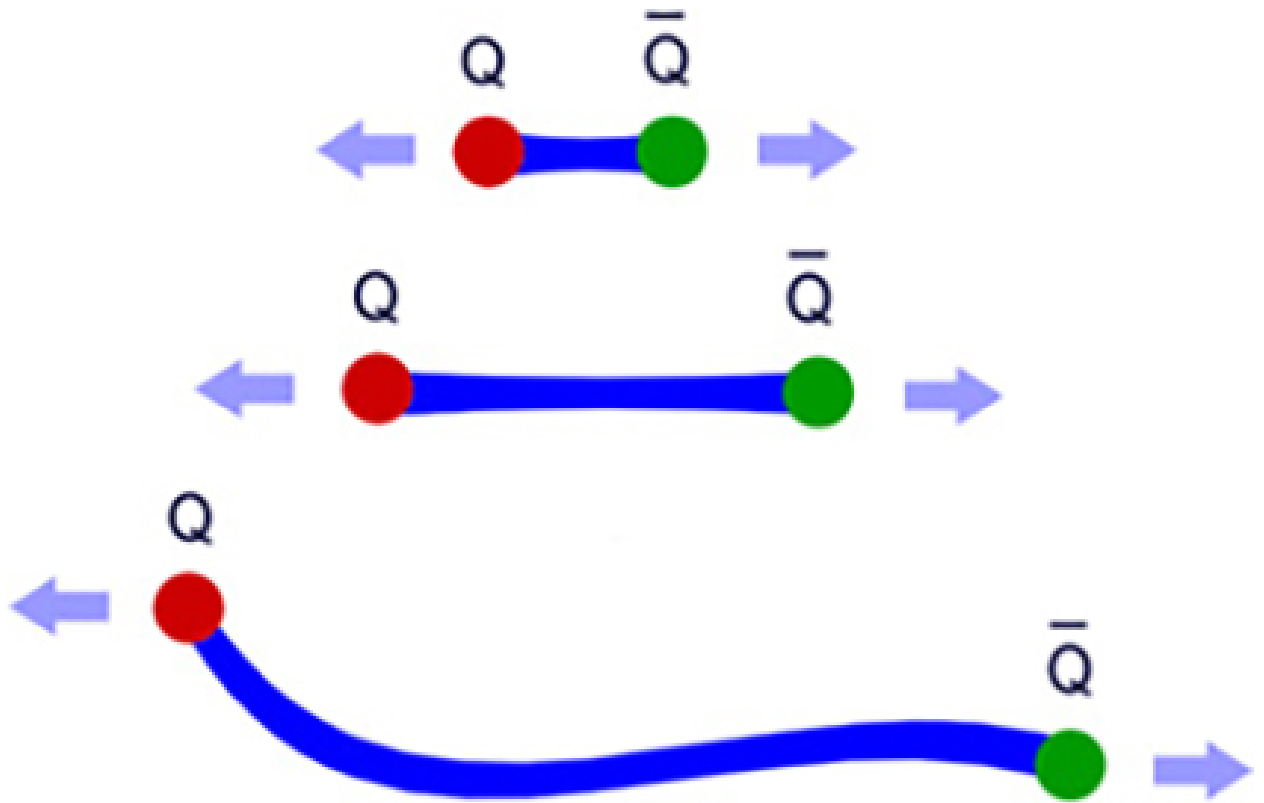


Рисунок 2.2 — Кварки

проявляться либо как рождение и распад частицы-переносчика новых сил (такие гипотетические частицы обозначают Z'), либо как усиленное рассеяние частиц на большие углы.

Лептокварки. В Стандартной модели и в подавляющем большинстве теорий Новой физики кварки и лептоны взаимодействуют друг с другом опосредованно, путем обмена квантами силовых полей. Однако можно представить себе возможность того, что кварки и лептоны исходно являлись фермионами одного типа и лишь потом расщепились на два разных сорта. В таком случае должны существовать новые тяжелые частицы — лептокварки, которые распадаются прямо на кварк и лептон. Подобные частицы встречаются в теориях Великого объединения.

Квирки. Одним из очень необычных и любопытных вариантов новых сил является гипотеза квирков (*quirks*). Эта модель построена по типу обычного сильного взаимодействия: в ней предполагается, что существует новое силовое поле с конфайнментом и новые частицы, его чувствующие. Если частицы очень тяжелые, то между ними будут натягиваться длинные, даже макроскопические силовые струны, которые не смогут порваться (рисунок 2.2).

Слабое взаимодействие — короткодействующее фундаментальное взаимодействие между элементарными частицами, ответственное за бета-

распад атомных ядер и медленные распады частиц. Слабое взаимодействие значительно слабее сильного и электромагнитного, но гораздо сильнее гравитационного. В слабом взаимодействии участвуют все фундаментальные фермионы (кварки и лептоны) и все адроны. Единственными частицами, которые участвуют только в слабом взаимодействии являются три типа нейтрино ν_e , ν_μ , ν_τ и их античастицы антинейтрино $\bar{\nu}_e$, антинейтрино $\bar{\nu}_\mu$, антинейтрино $\bar{\nu}_\tau$. В нем не участвуют переносчики сильного, электромагнитного и гравитационного взаимодействий – глюон, фотон и гравитон. В процессе слабого взаимодействия частицы обмениваются переносчиками слабого взаимодействия промежуточными (фундаментальными) бозонами: имеющими электрический заряд W^\pm и нейтральным Z . Эти бозоны, в отличие от переносчиков остальных фундаментальных сил безмассовых глюона, фотона и гравитона, имеют огромные массы $m_W = 80.4 \text{ ГэВ}/c^2$ и $m_Z = 91.2 \text{ ГэВ}/c^2$ (примерно как у атомов циркония или ниобия), что приводит к очень малому радиусу действия слабых сил $\approx 10^{-18} \text{ см}$ (что на три порядка меньше радиуса сильного взаимодействия) и очень низкой по сравнению с сильными и электромагнитными процессами вероятности (скорости) слабых процессов.

Несмотря на малую величину и короткодействие слабые силы играют очень важную роль в природе. Так без них погасло бы Солнце, так как внутри него остановился бы процесс превращения 4 протонов в ядро гелия-4, являющийся основным источником энергии Солнца.

Слабое взаимодействие выделяется тем, что в нём не соблюдается ряд запретов, присущих сильному и электромагнитному взаимодействиям. Так в слабых процессах кварки одного типа (аромата) превращаются в кварки других ароматов [13].

Особенности слабого взаимодействия

- Их слабость (медленность), выражающаяся в том, что вероятность этих процессов на много порядков меньше вероятностей сильных и электромагнитных процессов.
- Малый радиус взаимодействия — как минимум на два порядка меньший, чем радиус сильного взаимодействия. Ни в одном из слабых процессов не удалось до 1982 г. обнаружить каких-либо отклонений от точечного четырех- фермионного взаимодействия.
- Сильное, максимально возможное несохранение пространственной и зарядовой четностей. Так, в заряженные токи входят только левые компоненты спиноров, описывающих частицы, и только правые компоненты спиноров, описывающих античастицы.

- Несохранение CP -четности.
- Несохранение ароматов (странности, чарма и т. д.).
- То обстоятельство, что только в слабых взаимодействиях принимают участие нейтрино.

Тем поразительней, что, несмотря на столь резкие отличия, слабые и электромагнитные взаимодействия представляют собой, по-видимому, проявление одного и того же взаимодействия, которое в последние годы получило название электрослабого.

Согласно электрослабой теории слабые взаимодействия заряженных токов обусловлены обменами W -бозонами, а нейтральных – Z -бозонами, подобно тому как взаимодействие электромагнитных токов обусловлено обменом фотонами. При этом слабость и малый радиус слабого взаимодействия объясняются тем, что, в отличие от фотонов, W и Z -бозоны – очень тяжелые частицы. Остальные особенности слабого взаимодействия прямо заложены в предположении о форме исходных фермионных токов теории. Так что в электрослабой теории удивляться надо не тому, что слабое взаимодействие зеркально-асимметрично, а тому, что электромагнитное – зеркально-симметричное.

Слабое взаимодействие переносится массивными W^\pm - и Z -бозонами. Обмен заряженными W^+ и W^- -бозонами приводит к изменению электрического заряда взаимодействующих фермионов. Эти процессы происходят за счет заряженных токов.

В физических программах экспериментов на современных дронных (LHC) и планируемых на электрон-позитронных (ILC , $CLIC$) коллайдерах вопросу поиска «новой» физики, выходящей за рамки Стандартной модели (СМ), традиционно уделяется большое внимание. К числу подобных теоретических построений, являющихся обобщением СМ, относятся модели с расширенным калибровочным сектором, такие как лево-правосимметричные модели (LR), альтернативные лево-правосимметричные модели (ALR), E_6 -модели и др. [12]. Их исследование (теоретическое и экспериментальное) представляет значительный интерес. Эти модели являются одними из простейших расширений СМ, характеризующихся элементарной структурой хиггсовского сектора. Общим для данных моделей является то, что они предсказывают новые физические объекты и явления на масштабе энергий $O(1 \text{ ТэВ})$, связанные, например, с наличием тяжелых нейтральных (Z') калибровочных бозонов, обусловленных дополнительными калибровочными симметриями $U(1)'$.

Достижение порога рождения Z' -бозона явилось бы прямым доказательством проявления «новой» физики. Однако в данном случае интервал по-

иска масс Z' ограничен максимальной энергией коллайдера, на котором проводятся эксперименты. Значительно более широкий интервал масс можно исследовать с помощью пропагаторных эффектов. В этом случае ведется поиск отклонений различных наблюдаемых от соответствующих предсказаний СМ. Если экспериментальные данные при достигнутом уровне точности согласуются с СМ, т. е. отклонений от предсказаний СМ нет, то эту экспериментальную информацию можно использовать для получения ограничений на динамические параметры и массы Z' -бозонов.

Потенциальные возможности e^+e^- -коллайдеров для прямого рождения новых калибровочных бозонов гораздо скромнее по сравнению с адронными машинами из-за более низких энергий пучков. Кроме того, современные ограничения на массы Z' -бозонов для большинства моделей превосходят планируемую энергию электрон-позитронного коллайдера ILC , $\sqrt{s} \ll M_{Z'}$. Тем не менее основным достоинством этих машин является возможность проведения экспериментов по измерению наблюдаемых величин с высокой степенью точности и получения однозначной информации о косвенных (виртуальных) эффектах новых Z' -бозонов, а также эффектах бозонного Z - Z' -смешивания. Последние, в моделях с расширенным калибровочным сектором, зависят от структуры хиггсовского сектора модели. Тем самым экспериментальное исследование процессов рождения пар W^\pm -бозонов может не только пролить свет на возможное существование «новой» физики, но и дать косвенные указания на хиггсовскую природу, а также установить структуру модели.

На основе данных, полученных из низкоэнергетических экспериментов по нейтральным токам, результатов на e^+e^- -коллайдерах LEP и SLC [12], а также недавно выполненных экспериментов по поиску прямого адронного рождения Z' -бозонов в процессе Дрелла-Яна.

$$pp \rightarrow Z' \rightarrow l^+l^- + X$$

($l = e, \mu$) на коллайдере LHC при энергии $\sqrt{s} = 7$ и 8 ТэВ с интегральной светимостью соответственно $L_{int} = 5$ и 20 фб $^{-1}$ [12] можно заключить, что для большинства расширенных калибровочных моделей граничные значения для масс дополнительных Z' -бозонов находятся в интервале $\sim 2,5$ -3,0 ТэВ (в зависимости от модели), а современный масштаб ограничений на угол смешивания составляет $\mathcal{O}(\varphi) \sim 10^{-2}$ - 10^{-3} рад. При этом наиболее точная информация об угле смешивания была получена преимущественно из экспериментов на электрон-позитронных коллайдерах $LEP1$ [18] и SLC по измерению резонансных наблюдаемых физических величин при энергии начальных состояний, равной массе стандартного Z -бозона, $\sqrt{s} = M_Z$, в процессах

$$e^+e^- \rightarrow f\bar{f}$$

где конечными фермионными состояниями f были заряженные лептоны и кварки [17]. Высокая точность, достигнутая в экспериментах на коллайдерах *LEP1* и *SLC*, объясняется прежде всего возможностью набора большого объема данных в резонансной области энергии.

Кроме того, эта информация дополнялась данными, полученными на коллайдере тэватрон, по точному измерению массы M_W , на основе которых определялся параметр бозонного $Z-Z'$ -смешивания с использованием соотношения между массами нейтральных и заряженных калибровочных бозонов, $M_Z = M_W/(\sqrt{p_0} \cos \theta_W)$, имеющего место в расширенных моделях. Очевидно также, что эти данные будут дополнены новой информацией, которая в ближайшем будущем будет получена в экспериментах на коллайдере *LHC* при энергии 13 и 14 ТэВ. Вместе с тем из этих данных нельзя сделать однозначный вывод о природе «новой» физики, который мог бы вызвать отклонение наблюдаемых величин от их поведения, предсказываемого СМ. Дело в том, что параметр p , который содержится в выражениях для векторных и аксиально-векторных констант связи фермионов с учетом петлевых поправок, зависит, в частности, от структуры хиггсовского сектора модели, которая изначально неизвестна. Кроме того, новые тяжелые фермионы и скалярные частицы, предсказываемые моделями с расширенным калибровочным сектором, могут давать вклад в параметр p на петлевом уровне. Все эти неопределенности приводят к появлению систематических (теоретических) погрешностей, которые могут быть весьма существенными при измерении параметра p и, в конечном счете, могут повлиять на точность определения параметра $Z-Z'$ -смешивания.

Процессы парного рождения заряженных W^\pm -бозонов в адронных столкновениях на *LHC*

$$pp \rightarrow W^+W^- + X$$

электрон-позитронной аннигиляции на *LEP2* и в большей степени на *ILC*

$$e^+e^- \rightarrow W^+W^-$$

Являются весьма эффективным инструментом поиска эффектов $Z-Z'$ -смешивания при высоких энергиях и, таким образом, играют роль основного поставщика информации об угле $Z-Z'$ -смешивания [12]. С теоретической точки зрения процессы парного рождения заряженных калибровочных бозонов в адронных и электронпозитронных столкновениях интересны тем, что их сечения пропорциональны углу $Z-Z'$ -смешивания, который, как отмечалось выше, в расширенных калибровочных моделях зависит от структуры хиггсовского сектора [19].

Прямой поиск тяжелых резонансов в процессе $p\bar{p} \rightarrow W^+W^- + X$ осуществлялся экспериментальными группами *CDF* и *D0* на коллайдере тэватрон. Коллаборация *D0* исследовала возможность рождения резонанса в канале его дибозонного распада, используя чисто лептонные $lv'l'v'$ и полулептонные vjj моды. Здесь $l = e, \mu$; jj — две адронные струи. Коллаборация *CDF* также осуществляла поиск тяжелых резонансов в канале их распада в пару заряженных калибровочных бозонов W^+W^- с последующим распадом в полулептонные $evjj$ конечные состояния. Обе коллаборации установили ограничения на массы тяжелых резонансов, таких как новые нейтральные Z' - и заряженные калибровочные W^\pm -бозоны, гравитоны Рэндалл-Сандрума. Кроме того, в настоящее время поиск тяжелых резонансов на *LHC* в WW -канале интенсивно ведется коллаборациями *ATLAS* и *CMS*. В частности, уже получена экспериментальная информация о процессе в лептонном канале $lv'l'v'$ при энергии коллайдера 7 ТэВ и интегральной светимости $4,7 \text{ фб}^{-1}$ [16].

Из анализа экспериментальных данных по измерению процесса электрон-позитронной аннигиляции на коллайдере *LEP2* были впервые получены прямые ограничения на угол $Z-Z'$ -смешивания. Точность измерения угла смешивания оказалась не очень высокой, $|\phi|$ 5–10 %, так как сам коллайдер работал в интервале энергий, незначительно превышающем порог реакции, $\sqrt{s} \gg 2M_W$. Как было установлено ранее, чувствительность процесса электрон-позитронной аннигиляции к эффектам «новой» физики значительно усиливается при высоких энергиях, $\sqrt{s} \gg 2M_W$, где важную роль играет механизм калибровочного сокращения. Дело в том, что вклад Z' -бозона в сечение процесса нарушает механизм калибровочного сокращения, играющий важную роль в СМ. Действие механизма калибровочного сокращения состоит в том, что он обеспечивает «правильное» поведение сечения процесса электрон-позитронной аннигиляции с ростом энергии, которое не нарушает унитарный предел, несмотря на быстро растущие с энергией отдельные вклады в сечение. Вместе с тем эффекты, индуцированные появлением дополнительного калибровочного бозона, нарушают механизм калибровочного сокращения в энергетическом интервале $2M_W \ll \sqrt{s} \ll M_{Z'}$, что проявляется в виде «разбалансировки» отдельных вкладов в сечение и, как следствие, в возникновении существенно иной по сравнению со СМ энергетической зависимостью сечений. Этим обусловлено действие так называемого механизма усиления эффектов «новой» физики в процессе электрон-позитронной аннигиляции. Именно в силу этого обстоятельства линейный коллайдер *ILC* является одним из основных инструментариев для поиска эффектов «новой» физики при исследовании процесса электрон-позитронной аннигиляции.

Следует отметить также, что коллаборация *CDF* на коллайдере тэватрон одной из первых получила прямые ограничения на угол $Z-Z'$ -смешивания из обработки данных по измерению процесса адронного рождения W^+W^- -бозонов. И вновь относительно небольшая энергия установки и низкая светимость не позволили улучшить ограничения, полученные на коллайдере *LEP2*, а лишь повторить их [20].

Возможности коллайдера *LHC* по обнаружению эффектов $Z-Z'$ -смешивания в процессе рождения пар заряженных калибровочных W^\pm -бозонов с их последующим распадом по чисто лептонному каналу $lv'l'v'$. Несмотря на очевидное достоинство данного канала, связанное с подавленностью фона, особенно при больших инвариантных массах W^\pm -бозонов, у него имеется заметный недостаток, связанный с присутствием в конечных фермионных состояниях двух нейтрино, что не позволяет восстановить распределение по инвариантной массе бозонных пар из экспериментальных данных. В то же время распад пары W^\pm -бозонов по полулептонному каналу $lvjj$ свободен от указанного недостатка. В процессе $pp \rightarrow Z' \rightarrow WW + X \rightarrow lvjj + X$ существует возможность реконструировать распределение по инвариантной массе W^+W^- - пары и тем самым исследовать резонансную структуру Z' -бозона. Еще одним достоинством настоящего полулептонного процесса является то, что он имеет сечение, существенно превосходящее сечение чисто лептонного канала. Вместе с тем полулептонный канал, в отличие от лептонного канала $lv'l'v'$, имеет большой КХД-фон, вызванный рождением W_{jj^-} , а также Z_{jj} -состояний [21]. В последнем случае предполагается, что Z -бозон распадается по лептонному каналу, а в процессе детектирования лептонов один из них теряется. Кроме перечисленных выше КХД фоновых процессов имеется еще один, который играет важную роль в оценке всей фоновой составляющей. Это процесс рождения пар $t\bar{t}$ -кварков. Однако большой КХД-фон может быть редуцирован путем наложения кинематических ограничений на поперечные импульсы заряженных лептонов и адронных струй в резонансном сигнале рождения Z' -бозонов [12].

2.2 Используемые средства разработки программного обеспечения

Проект реализован посредством языка программирования *Java*. *Java* – объектно-ориентированный язык программирования, разработанный компанией *Sun Microsystems* (в последующем приобретённой компанией *Oracle*). Приложения *Java* обычно транслируются в специальный байт-код, поэтому они могут работать на любой виртуальной Java-машине вне зави-

симости от компьютерной архитектуры. Дата официального выпуска – 23 мая 1995 года [4]. Программы на *Java* транслируются в байт-код, выполняемый виртуальной машиной *Java (JVM)* – программой, обрабатывающей байтовый код и передающей инструкции оборудованию как интерпретатор. Достоинством подобного способа выполнения программ является полная независимость байт-кода от операционной системы и оборудования, что позволяет выполнять Java-приложения на любом устройстве, для которого существует соответствующая виртуальная машина. Другой важной особенностью технологии *Java* является гибкая система безопасности, в рамках которой исполнение программы полностью контролируется виртуальной машиной. Любые операции, которые превышают установленные полномочия программы (например, попытка несанкционированного доступа к данным или соединения с другим компьютером), вызывают немедленное прерывание. Часто к недостаткам концепции виртуальной машины относят снижение производительности. Ряд усовершенствований несколько увеличил скорость выполнения программ на *Java*:

1. Применение технологии трансляции байт-кода в машинный код непосредственно во время работы программы (*JIT*-технология) с возможностью сохранения версий класса в машинном коде;
2. Широкое использование платформенно-ориентированного кода (*native*-код) в стандартных библиотеках;
3. Аппаратные средства, обеспечивающие ускоренную обработку байт-кода (например, технология *Jazelle*, поддерживаемая некоторыми процессорами фирмы *ARM*).

Для семи разных задач время выполнения на *Java* составляет в среднем в полтора-два раза больше, чем для *C/C++*, в некоторых случаях *Java* быстрее, а в отдельных случаях в 7 раз медленнее. С другой стороны, для большинства из них потребление памяти *Java*-машиной было в 10–30 раз больше, чем программой на *C/C++*. Также примечательно исследование, проведённое компанией *Google*, согласно которому отмечается существенно более низкая производительность и большее потребление памяти в тестовых примерах на *Java* в сравнении с аналогичными программами на *C++* [4]. Идеи, заложенные в концепцию и различные реализации среды виртуальной машины *Java*, вдохновили множество энтузиастов на расширение перечня языков, которые могли бы быть использованы для создания программ, исполняемых на виртуальной машине. Эти идеи нашли также выражение в спецификации общезыковой инфраструктуры *CLI*, заложенной в основу платформы. Внутри *Java* существуют несколько основных семейств технологий:

1. *Java SE – Java Standard Edition*, основное издание *Java*, содержит компиляторы, *API*, *Java Runtime Environment* подходит для создания пользовательских приложений, в первую очередь – для настольных систем.
2. *Java EE – Java Enterprise Edition*, представляет собой набор спецификаций для создания программного обеспечения уровня предприятия.
3. *Java ME – Java Micro Edition*, создана для использования в устройствах, ограниченных по вычислительной мощности, например, в мобильных телефонах, КПК, встроенных системах;
4. *JavaFX* – технология, являющаяся следующим шагом в эволюции *Java* как *Rich Client Platform*, которая предназначена для создания графических интерфейсов корпоративных приложений и бизнеса.
5. *Java Card* – технология предоставляет безопасную среду для приложений, работающих на смарт-картах и других устройствах с очень ограниченным объёмом памяти и возможностями обработки.

Следующие успешные проекты реализованы с привлечением *Java (J2EE)* технологий: «*RuneScape*», «*Amazon*», «*eBay*», «*LinkedIn*», «*Yahoo!*».

Следующие компании в основном фокусируются на *Java (J2EE)* технологиях: *SAP*, *IBM*, *Oracle*. В частности, СУБД *Oracle Database* включает *JVM* как свою составную часть, обеспечивающую возможность непосредственного программирования СУБД на языке *Java*, включая, например, хранимые процедуры [4].

Программы, написанные на *Java*, имеют репутацию более медленных и занимающих больше оперативной памяти, чем написанные на языке *C*. Тем не менее, скорость выполнения программ, написанных на языке *Java*, была существенно улучшена с выпуском в 1997–1998 годах так называемого *JIT*-компилятора в версии 1.1 в дополнение к другим особенностям языка для поддержки лучшего анализа кода (такие, как внутренние классы, класс *StringBuffer*, упрощенные логические вычисления и т. д.). Кроме того, была произведена оптимизация виртуальной машины *Java* – с 2000 года для этого используется виртуальная машина *HotSpot*. По состоянию на февраль 2012 года, код *Java 7* приблизительно в 1.8 раза медленнее кода, написанного на языке *C*.

Некоторые платформы предлагают аппаратную поддержку выполнения для *Java*. К примеру, микроконтроллеры, выполняющие код *Java* на аппаратном обеспечении вместо программной *JVM*, а также основанные на *ARM* процессоры, которые поддерживают выполнение байткода *Java* через опцию *Jazelle*.

Основные возможности:

1. Автоматическое управление памятью;
2. Расширенные возможности обработки исключительных ситуаций;
3. Богатый набор средств фильтрации ввода-вывода;
4. Набор стандартных коллекций: массив, список, стек и т. п.;
5. Наличие простых средств создания сетевых приложений (в том числе с использованием протокола *RMI*);
6. Наличие классов, позволяющих выполнять *HTTP*-запросы и обрабатывать ответы;
7. Встроенные в язык средства создания многопоточных приложений, которые потом были портированы на многие языки (например, *python*);
8. Унифицированный доступ к базам данных на уровне отдельных *SQL*-запросов – на основе *JDBC*, *SQLJ*;
9. Поддержка обобщений (начиная с версии 1.5);
10. Поддержка лямбд, замыканий, встроенные возможности функционального программирования (с версии 1.8);
11. Наличие вариантов реализации многопоточных программ

Разработчику на *Java* доступно множество готовых (или библиотечных) классов и методов, полезных для использования в собственных программах. Наличие библиотечных решений позволяет изящно решать множество задач. Рассматриваемый компонент позволит преобразовать вычисления в программный код.

Spring Framework (или коротко *Spring*) — универсальный фреймворк с открытым исходным кодом для Java-платформы.

Первая версия была написана Родом Джонсоном, который впервые опубликовал её вместе с изданием своей книги «*Expert One-on-One Java EE Design and Development*» [5] (*Wrox Press*, октябрь 2002 года).

Фреймворк был впервые выпущен под лицензией *Apache 2.0 license* в июне 2003 года. Первая стабильная версия 1.0 была выпущена в марте 2004. *Spring 2.0* был выпущен в октябре 2006, *Spring 2.5* — в ноябре 2007, *Spring 3.0* в декабре 2009, и *Spring 3.1* в декабре 2011. Текущая версия — 5.1.2 [5].

Несмотря на то, что *Spring* не обеспечивал какую-либо конкретную модель программирования, он стал широко распространённым в *Java*-сообществе главным образом как альтернатива и замена модели *Enterprise JavaBeans*. *Spring* предоставляет большую свободу *Java*-разработчикам в проектировании. Кроме того, он предоставляет хорошо документированные и лёгкие в использовании средства решения проблем, возникающих при создании приложений корпоративного масштаба.

Между тем, особенности ядра *Spring* применимы в любом *Java*-приложении, и существует множество расширений и усовершенствований для построения веб-приложений на *Java Enterprise* платформе. По этим причинам *Spring* приобрёл большую популярность и признаётся разработчиками, как стратегически важный фреймворк.

Spring обеспечивает решения многих задач, с которыми сталкиваются *Java*-разработчики и организации, которые хотят создать информационную систему, основанную на платформе *Java*. Из-за широкой функциональности трудно определить наиболее значимые структурные элементы, из которых он состоит. *Spring* не всецело связан с платформой *Java Enterprise*, несмотря на его масштабную интеграцию с ней, что является важной причиной его популярности.

Spring, вероятно, наиболее известен как источник расширений (*features*), нужных для эффективной разработки сложных бизнес-приложений вне тяжёловесных программных моделей, которые исторически были доминирующими в промышленности. Ещё одно его достоинство в том, что он ввел ранее неиспользуемые функциональные возможности в сегодняшние господствующие методы разработки, даже вне платформы *Java*.

Этот фреймворк предлагает последовательную модель и делает её применимой к большинству типов приложений, которые уже созданы на основе платформы *Java*. Считается, что *Spring* реализует модель разработки, основанную на лучших стандартах индустрии, и делает её доступной во многих областях *Java*.

Spring может быть рассмотрен как коллекция меньших фреймворков или фреймворков во фреймворке. Большинство этих фреймворков может работать независимо друг от друга, однако они обеспечивают большую функциональность при совместном их использовании. Эти фреймворки делятся на структурные элементы типовых комплексных приложений:

1. *Inversion of Control*-контейнер: конфигурирование компонентов приложений и управление жизненным циклом *Java*-объектов.
2. Фреймворк аспектно-ориентированного программирования: работает с функциональностью, которая не может быть реализована возмож-

ностями объектно-ориентированного программирования на *Java* без потерь.

3. Фреймворк доступа к данным: работает с системами управления реляционными базами данных на *Java*-платформе, используя *JDBC*- и *ORM*-средства и обеспечивая решения задач, которые повторяются в большом числе *Java-based environments*.
4. Фреймворк управления транзакциями: координация различных *API* управления транзакциями и инструментарий настраиваемого управления транзакциями для объектов *Java*.
5. Фреймворк *MVC*: каркас, основанный на *HTTP* и сервлетах, предоставляющий множество возможностей для расширения и настройки (*customization*).
6. Фреймворк удалённого доступа: конфигурируемая передача *Java*-объектов через сеть в стиле *RPC*, поддерживающая *RMI*, *CORBA*, *HTTP-based* протоколы, включая *web*-сервисы (*SOAP*).
7. Фреймворк аутентификации и авторизации: конфигурируемый инструментарий процессов аутентификации и авторизации, поддерживающий много популярных и ставших индустриальными стандартами протоколов, инструментов, практик через дочерний проект *Spring Security* (ранее известный как *Acegi*).
8. Фреймворк удалённого управления: конфигурируемое представление и управление *Java*-объектами для локальной или удалённой конфигурации с помощью *JMX*.
9. Фреймворк работы с сообщениями: конфигурируемая регистрация объектов-слушателей сообщений для прозрачной обработки сообщений из очереди сообщений с помощью *JMS*, улучшенная отправка сообщений по стандарту *JMS API*.
10. Тестирование: каркас, поддерживающий классы для написания модульных и интеграционных тестов.

Центральной частью *Spring* является контейнер *Inversion of Control*, который предоставляет средства конфигурирования и управления объектами *Java* с помощью рефлексии. Контейнер отвечает за управление жизненным циклом объекта: создание объектов, вызов методов инициализации и конфигурирование объектов путём связывания их между собой.

Объекты, создаваемые контейнером, также называются управляемыми объектами (*beans*). Обычно конфигурирование контейнера осуществляется

путём загрузки *XML*-файлов, содержащих определение *bean*'ов и предоставляющих информацию, необходимую для создания *bean*'ов.

Spring имеет собственную *MVC*-платформу веб-приложений, которая не была первоначально запланирована. Разработчики *Spring* решили написать её как реакцию на то, что они восприняли как неудачность конструкции (тогда) популярного Apache *Struts*, а также других доступных веб-фреймворков. В частности, по их мнению, было недостаточным разделение между слоями представления и обработки запросов, а также между слоем обработки запросов и моделью.

Класс *DispatcherServlet* является основным контроллером фреймворка и отвечает за делегирование управления различным интерфейсам, на всех этапах выполнения *HTTP*-запроса. Об этих интерфейсах следует сказать более подробно.

Как и *Struts*, *Spring MVC* является фреймворком, ориентированным на запросы. В нем определены стратегические интерфейсы для всех функций современной запросно-ориентированной системы. Цель каждого интерфейса — быть простым и ясным, чтобы пользователям было легко его заново имплементировать, если они того пожелают. *MVC* прокладывает путь к более чистому *front-end*-коду. Все интерфейсы тесно связаны с *Servlet API*. Эта связь рассматривается некоторыми как неспособность разработчиков Spring предложить для веб-приложений абстракцию более высокого уровня. Однако эта связь оставляет особенности *Servlet API* доступными для разработчиков, облегчая все же работу с ним.

Docker — программное обеспечение для автоматизации развёртывания и управления приложениями в средах с поддержкой контейнеризации. Позволяет «упаковать» приложение со всем его окружением и зависимостями в контейнер, который может быть перенесён на любую *Linux*-систему с поддержкой *cgroups* в ядре, а также предоставляет среду по управлению контейнерами. Изначально использовал возможности *LXC*, с 2015 года применял собственную библиотеку, абстрагирующую виртуализационные возможности ядра *Linux* — *libcontainer*. С появлением *Open Container Initiative* начался переход от монолитной к модульной архитектуре [6].

Программное обеспечение функционирует в среде *Linux* с ядром, поддерживающим *cgroups* и изоляцию пространств имён (*namespaces*); существуют сборки только для платформ *x86-64* и *ARM* [6]. Начиная с версии 1.6 возможно использование в ОС *Windows*.

Для экономии дискового пространства проект использует файловую систему *Aufs* с поддержкой технологии каскадно-объединённого монтирования: контейнеры используют образ базовой операционной системы, а изменения записываются в отдельную область. Также поддерживается раз-

мещение контейнеров в файловой системе *Btrfs* с включённым режимом копирования при записи.

В состав программных средств входит демон — сервер контейнеров, клиентские средства, позволяющие из интерфейса командной строки управлять образами и контейнерами, а также *API*, позволяющий в стиле *REST* управлять контейнерами программно.

Демон обеспечивает полную изоляцию запускаемых на узле контейнеров на уровне файловой системы (у каждого контейнера собственная корневая файловая система), на уровне процессов (процессы имеют доступ только к собственной файловой системе контейнера, а ресурсы разделены средствами *libcontainer*), на уровне сети (каждый контейнер имеет доступ только к привязанному к нему сетевому пространству имён и соответствующим виртуальным сетевым интерфейсам).

Набор клиентских средств позволяет запускать процессы в новых контейнерах (*docker run*), останавливать и запускать контейнеры (*docker stop* и *docker start*), приостанавливать и возобновлять процессы в контейнерах (*docker pause* и *docker unpause*). Серия команд позволяет осуществлять мониторинг запущенных процессов (*docker ps* по аналогии с *ps* в *Unix*-системах, *docker top* по аналогии с *top* и другие). Новые образы возможно создавать из специального сценарного файла (*docker build*, файл сценария носит название *Dockerfile*), возможно записать все изменения, сделанные в контейнере, в новый образ (*docker commit*). Все команды могут работать как с *docker*-демоном локальной системы, так и с любым сервером *Docker*, доступным по сети. Кроме того, в интерфейсе командной строки встроены возможности по взаимодействию с публичным репозиторием *Docker Hub*, в котором размещены предварительно собранные образы контейнеров, например, команда *docker search* позволяет осуществить поиск образов среди размещённых в нём, образы можно скачивать в локальную систему (*docker pull*), возможно также отправить локально собранные образы в *Docker Hub* (*docker push*) [6].

Также *Docker* имеет пакетный менеджер *Docker Compose*, позволяющий описывать и запускать многоконтейнерные приложения. Конфигурационные файлы *Compose* описываются на языке *YAML*.

Amazon Web Services (AWS) – наиболее распространенная в мире облачная платформа с самыми широкими возможностями, которая предоставляет 165 полнофункциональных сервисов для центров обработки данных по всей планете. Миллионы клиентов, в том числе стартапы, ставшие лидерами по скорости роста, крупнейшие корпорации и передовые правительственные учреждения, доверяют *AWS* в вопросах размещения инфраструктуры, повышения гибкости и снижения затрат [7].

AWS предоставляет сервисы для широкого спектра приложений, включая вычислительные сервисы, сервисы хранилищ, баз данных, сетевых конфигураций, аналитики, машинного обучения и искусственного интеллекта, Интернета вещей (*IoT*), обеспечения безопасности, сервисы для разработки и развертывания приложений, а также управления ими.

AWS обеспечивает не только самый большой спектр сервисов, но и самые широкие функциональные возможности в их рамках. Например, *Amazon EC2* предлагает больше типов и размеров вычислительных инстансов, чем любой другой поставщик, в том числе самые мощные инстансы с графическими процессорами для рабочих нагрузок, связанных с машинным обучением. *AWS* также обеспечивает вдвое больше сервисов баз данных, чем ближайшие конкуренты, и предлагает одиннадцать реляционных и нереляционных баз данных. К тому же, *AWS* обеспечивает больше всего способов запуска контейнеров: с помощью *Amazon Elastic Container Service (ECS)*, *Amazon Elastic Container Service for Kubernetes (EKS)* и *AWS Fargate* [7].

Широкий выбор сервисов и разнообразные функциональные возможности обеспечивают более простую, быструю и экономичную миграцию существующих приложений и предоставляют почти безграничные возможности для разработки.

ГЛАВА 3

РАЗРАБОТКА WEB-ПРИЛОЖЕНИЯ

В наши дни, когда компьютерные технологии бурно развиваются, не всегда удается создать сложное приложение, используя один язык программирования. Разные языки имеют свои преимущества и недостатки и как правило, что ни один из них не удовлетворят требованиям разрабатываемой прикладной программы. Выходом из такого положения является использование нескольких языков программирования. Такой подход часто используется при создании программ для научных исследований, управления производственными процессами и других коммерческих приложений. При этом приходится решать задачи взаимодействия компонентов, написанных, которые написаны на на разных языках программирования. Компоненты, реализующие графический интерфейс пользователя, управление базами данных, получение и обработку данных в реальном времени, как правило интегрируются в одно приложение.

Для реализации поставленной задачи исследования в магистерской диссертации был выбран пакет моделирования процессов столкновения элементарных частиц при высоких энергиях на ускорителях элементарных частиц *PYTHIA* на языке программирования *C++*, а также принято решение о реализации всего программного комплекса на языке программирования *Java*. В качестве веб интерфейса был использован фреймворк *Angular* на языке программирования *JavaScript*. Так же была использована и библиотека *d3js* для отрисовки графиков в виде *SVG* изображений и взаимодействия пользователя с интерфейсом приложения.

Когда говорят о научных основах проектирования пользовательских интерфейсов, в первую очередь упоминают термин *Human-Computer Interaction (HCI)* – «взаимодействие человека и компьютера». В странах Запада *HCI* это является целой профессией, ей обучают в университетах, издается много журналов по этой теме, существует большое количество Web-сайтов. Составными частями *HCI* являются:

- человек (пользователь);
- компьютер;
- их взаимодействие.

Пользовательский интерфейс *user interface* (*UI*) – является своеобразным коммуникационным каналом, по которому осуществляется взаимодействие пользователя и компьютера.

Лучший пользовательский интерфейс – это такой интерфейс, которому пользователь не должен уделять много внимания, почти не замечать его. В руках пользователя интерфейс пользователя должен служить инструментом для достижения цели. Такой интерфейс называют прозрачным – пользователь смотрит сквозь него на свою работу.

Чтобы создать эффективный интерфейс, который делал бы работу с программным комплексом эффективной, нужно понимать, какие задачи будут решать пользователи с помощью данной программного комплекса и какие требования к интерфейсу могут возникнуть у пользователей. Большую роль в разработке интерфейса играет интуиция – если разработчик сам терпеть не может некрасивые и неудобные интерфейсы, то при создании собственного программного комплекса он будет чувствовать, где и какой именно элемент нужно убрать или добавить. Необходимо иметь художественный вкус, чтобы понимать, что именно придаст интерфейсу красоту и привлекательность.

Западные исследователи в области *HCI* сформулировали основные принципы проектирования пользовательских интерфейсов компьютерных программ [9]. Как и в любой другой отрасли ИТ, существует довольно много различных методик и классификаций. Можно сформировать три положения говоря об общих принципах проектирования пользовательского интерфейса:

- программный комплекс должен помогать выполнить задачу, а не становиться этой задачей;
- при работе с программой пользователь не должен думать, что он не понимает программу;
- программный комплекс должен работать так, чтобы пользователь не считал компьютер бесполезным инструментом.

Конечно, глубина проработки интерфейса и степень его адаптивности под нужды пользователя в программных комплексах в основном зависит от усилий их авторов, а не от характеристик аппаратного обеспечения. Однако у большинства пользователей компьютер ассоциируется именно с программными комплексами, которые на нем работают, и плохое впечатление от использования программного обеспечения автоматически переносится на сам компьютер.

Сообщество разработчиков фреймворка *Angular* разрабатывает дополнительные компоненты *Angular Material Design* и предлагает использовать их для быстрой разработки приложения с нуля.

Material Design – визуальный язык, представлен в 2014 году *Google*, используется чаще всего в мобильных приложениях. Пример использования *Material Design* можно увидеть во многих мобильных приложениях *Google* (*Play*, *Music*, *Books* и т.д.), а также в *Chrome OS*. *Material Design* упрощает разработчикам настройку *UI*, сохраняя при этом удобный интерфейс приложений. *Angular Material* состоит из набора предустановленных компонентов *Angular*. *Angular Material* стремится обеспечить расширенный и последовательный пользовательский интерфейс. В то же время он дает возможность контролировать, как ведут себя разные компоненты.

Открыв начальную веб-страницу приложения в любом из доступных браузеров пользователь увидит сообщение с описанием проекта, как показано на рисунке 3.1.

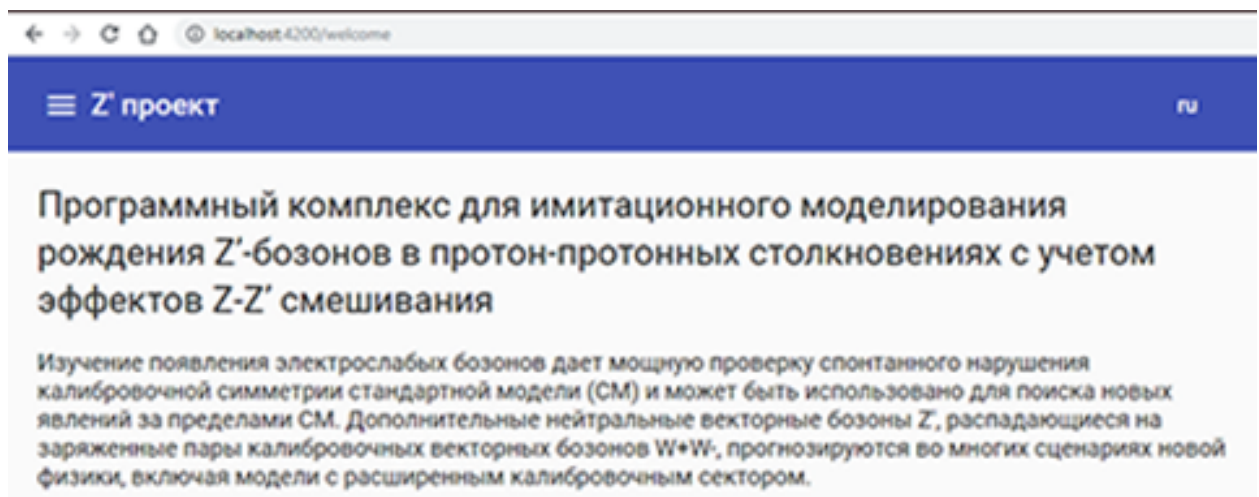


Рисунок 3.1 – Начальная *web*-страница

Для каждого пользователя на начальной странице приложения располагается навигационное меню, которое показано на рисунке 3.2. Так как разработанное приложение является одностраничным то переход по пунктам меню не перезагружает страницу полностью, а лишь догружает необходимые компоненты.

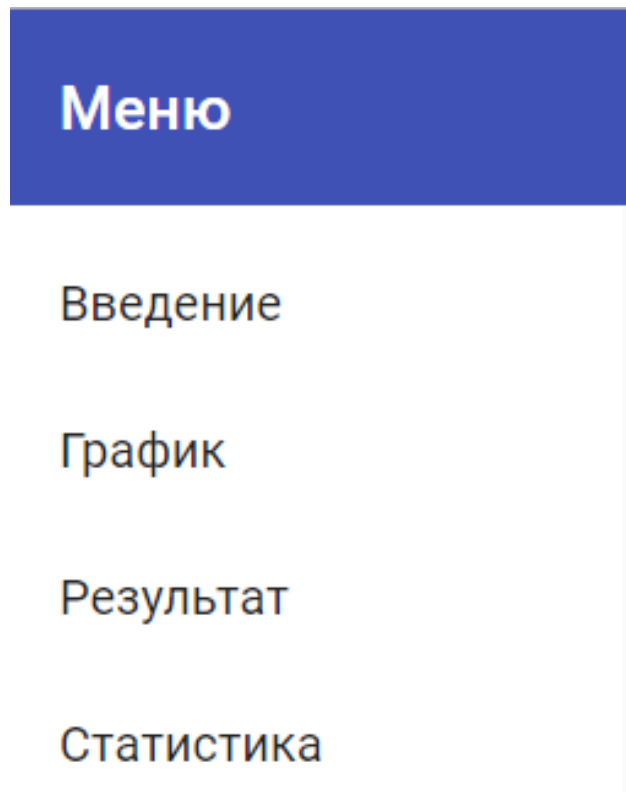


Рисунок 3.2 — Главное меню приложения

Из главного меню доступен переход на следующие страницы приложения:

- «Введение» – начальная страница *web*-приложения;
- «График» – страницы с основным графиком и панелью для ввода параметров и отправки запроса на вычисление;
- «Результат» – страница предоставляющая полученные результаты на угол смешивания Z' -бозонов в модели SSM ;
- «Статистика» – страница со статистикой приложения;

Перейдя на страницу «График» пользователь увидит пустой график распределения теоретического сечения кросс-секции $\sigma \times Br(Z' \rightarrow W^+W^-)$ для Z'_{SSM} и множество панелей управления, которые позволяют отправить запрос на начало эмуляции процесса $pp \rightarrow W^+W^- + X$ в протон-протонном столкновении.

Для начала старта генерации событий необходимо заполнить значение кси (ξ), количество моделируемых событий и количество циклов моделирования одного значения на графике для одного значения массы $M_{Z'}$. Данная панель показана на рисунке 3.3

Запросить на вычисление линию

Отправляет запрос на вычисление всей линии с заданными параметрами

Кси

Отправить

Кол-во моделируемых событий

☐ Отложенный

Кол-во циклов моделирования одной точки

Рисунок 3.3 — Панель старта расчета линии значений

После выставления нужных пользователю параметров, необходимо нажать кнопку «Отправить» после чего данные отправятся на сервер и результат будет перерисован на интерфейсе приложения. Результаты вычислений будут добавлены на основной график в реальном времени, поэтому пользователю не нужно обновлять страницу с графиком.

Как только запрос был отправлен на сервер создается *websocket* соединение между *web*-браузером пользователя и серверным приложением, тем самым подписывая клиента на получении данных от сервера, как только они будут готовы.

На графике кросс-секции $\sigma \times Br(Z' \rightarrow W^+W^-)$ для Z'_{SSM} отрисовывается линия с новыми значениями в то время, как серверное приложение начинает параллельно для нескольких значений массы $M_{Z'}$ моделировать столкновение протонов.

Так как в реальном столкновении протонов рождается очень большое количество различных частиц то моделирование такой задачи при помощи генератора *PYTHIA* требует значительных ресурсов центрального процессора сервера и занимает продолжительное время. Приложение созданное в рамках данной работы имеет *web*-интерфейс, что позволяет сразу нескольким пользователям работать в нем и отправлять запросы на вычисления. В связи с этим вводится ограничение на количество параллельно запущенных процессов моделирования для каждого пользователя.

Даже самая простая задача расчета модели занимает большое время. К примеру запрос пользователя рассчитать линию для всех значений $M_{Z'}$, а в рамках данного проекта это пять тысяч точек на графике от 0 до 5000 ГэВ, и количеством генерируемых событий займет 9,7 часа.

В таком простом случае полученное значение будет подвержено значительной статистической ошибки. Чтобы убрать данную статистическую ошибку требуются дополнительные циклы имитационного моделирования, что увеличивает затраченное время в разы.

Под панелью для запроса вычислений располагается список ранее отправленных запросов, как показано на рисунке 3.4. Нажав на крестик «X» можно удалить линию на графике.

Все запросы на вычисление сохраняются для конкретного пользователя и загружаются из локального хранилища данных при обновлении страницы.

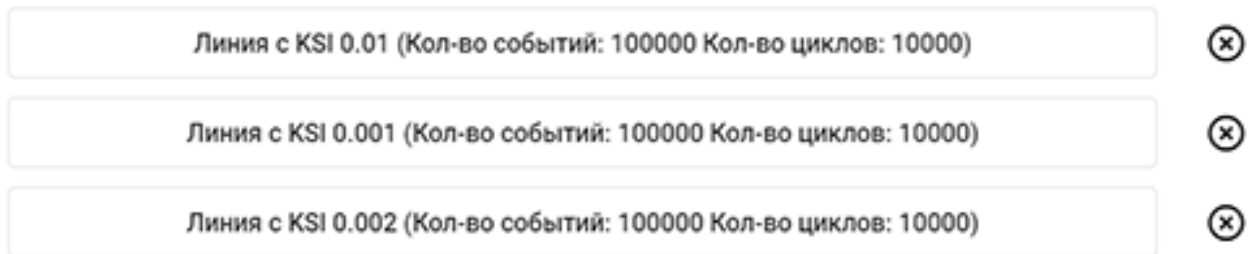


Рисунок 3.4 — Список отображаемых линий

Так как вычисления могут занять довольно много времени то вычисляются не все значения массы M_Z . Устанавливается шаг для вычислений к примеру 100 ГэВ и в качестве входного значения массы подставляются значения в 100 ГэВ, 200 ГэВ и т.д. Этот подход экономит значительное количество ресурсов вычислительной машины и экономит время для полусения общей картины поведения распределения ограничений на угол смешивания. Дополнительно на стороне сервера предусмотрено хранение уже рассчитанных результатов по выходным данным имметационного моделирования.

Как видно из рисунка 3.5 у пользователя есть возможно отправить «Отложенный» запрос на вычисление для удобства вычисления линий. Отложенный запрос означает, что запрос добавиться в очередь на вычислесе-ния на стороне сервера. Запрос будет сохраняться до того момента пока вычислительные ресурсы не освободятся и только после этого запрос обрабатывается и начнется процесс имметационного моделирования.

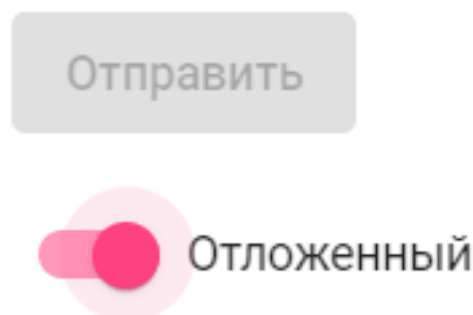


Рисунок 3.5 — Переключатель в «отложенный» режим отправки запросов

Предусмотрена возможность отправлять запросы на вычисление только одной точки на графике. Данная возможность позволяет узнать точное значение кросс-секции $\sigma \times Br(Z' \rightarrow W^+W^-)$ для конкретной массы Z'_{SSM} -бозона и не тратить время на вычисления значений всех масс $M_{Z'}$ для заданной ξ . Панель позволяющая отправлять такие запросы представлена на картинке 3.6.

Запросить точку

Отправляет запрос на вычисление точки

Кси

Отправить

Масса

Кол-во моделируемых событий

Кол-во циклов моделирования одной точки

Рисунок 3.6 — Панель старта расчета одного значения

Общий процесс вычисления всех значений для масс $M_{Z'}$ отображается в виде прогресс бара в самом низу веб страницы. Как показано на рисунке 3.7 данный элемент не имеет каких-либо точек взаимодействия с пользователем и несет лишь информативный характер.

Прогресс вычислений

В процессе вычислений

0

Лимит вычисляемых точек в одно время

10

Очередь запросов

0

Посчитано

0

Рисунок 3.7 — Прогресс вычислений

Компонент «Прогресс вычислений» содержит комплексную информацию о ходе вычислений: количество вычисляемых значений в данный момент времени, лимит вычисляемых точек в одно время для сессии одного пользователя, очередь запросов от пользователя и количество успешно полученных результатов.

Так как вычисления могут занять довольно много времени то вычисляются только текущие запросы для текущей сессии пользователя. К примеру если, лимит на вычисления равен десять запросов в одно время и пользователь запустил процесс вычисления целой линии значений, а после закрыл старницу *web*-приложения то будут произведены вычисления только этих десяти значений. Дальнейшие вычисления будут возобновлены, когда пользователь внонь откроет страницу *web*-интерфейса или другой пользователь отправить запрос на вычисление с теми же входными параметрами.

Интерфейс результатов и вычислений выполнен в виде *SVG* элемента и состоит из графика кросс-секции $\sigma \times Br(Z' \rightarrow W^+W^-)$ для масс $M_{Z'}$ (рисунок 3.8).

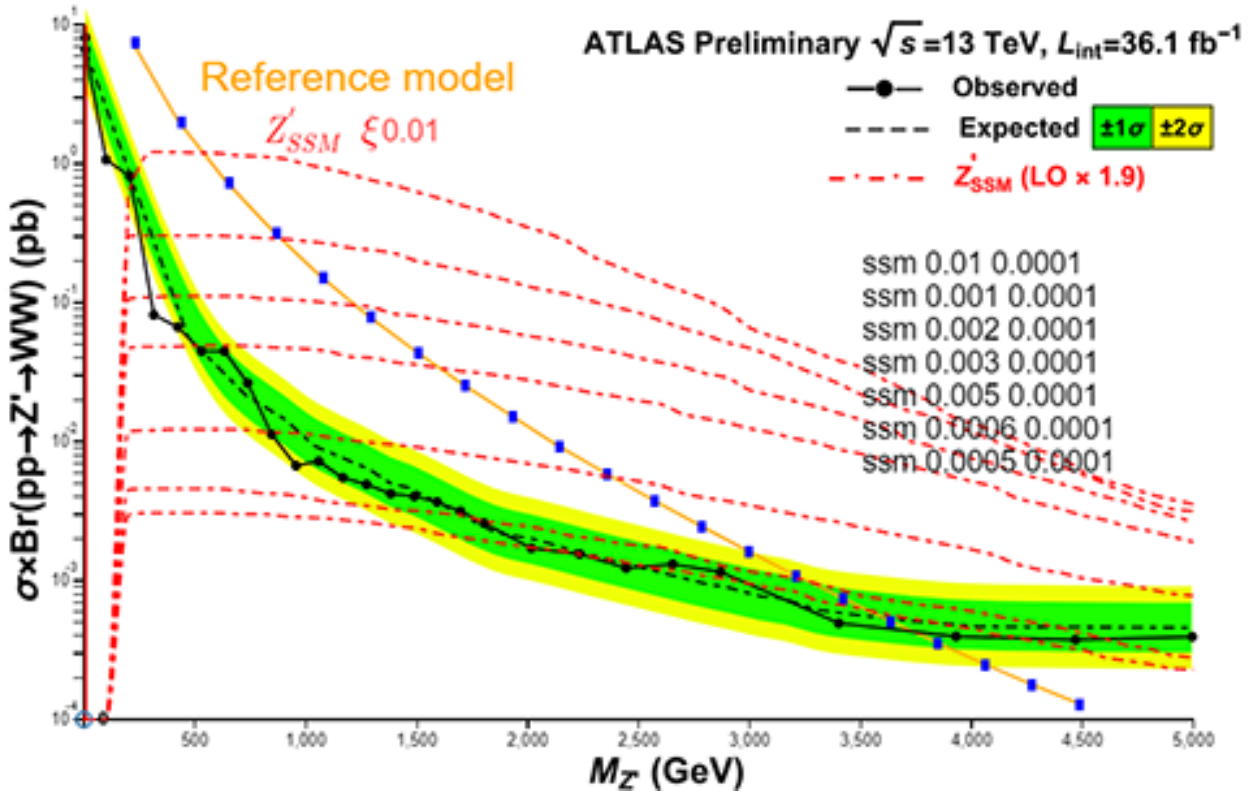


Рисунок 3.8 — Панель старта расчета линии значений

На графике отображены статические элементы полученные из математических формул и вычислений, а также данные не вычисляемые в рамках данной диссертации и полученные от коллаборации *ATLAS*, такие как *reference model*, *observed* значения, *expected* значения с областью в два стан-

дартных отклонения [16].

Представленный на рисунке 3.8 график имеет элемент управления - вертикальная красная линия. Данная линия добавлена, чтобы помочь пользователю определить значение кросс-секции для массы $M_{Z'}$ и выбранным углом смешивания ξ . Близлежащие значения к данной вертикальной линии будут подсвечены на графике и их точное значение всегда отображается в правой части графика.

Перейдя на вкладку «Результат» из меню приложения (рисунок 3.2) пользователю будет предоставлен результат работы приложения, который показан на рисунке 3.9.

Результат представляет собой график зависимости угла смешивания ξ и массы $M_{Z'}$ и отражает ограничения на угол смешивания Z' -бозонов в модели SSM .

Значениями ломаной *Observed* являются точки (ξ) пересечения моделируемых линий и кривой *expected*, которые отображены на рисунке 3.8. В данной диссертационной работе при помощи разработанной имитационной модели рождения Z' - бозонов в протон-протонных столкновениях с учетом эффектов $Z - Z'$ смешивания были получены ограничение на угол смешивания Z' -бозонов для до светимости 3000 fb^{-1} и равняется 6×10^{-5} .

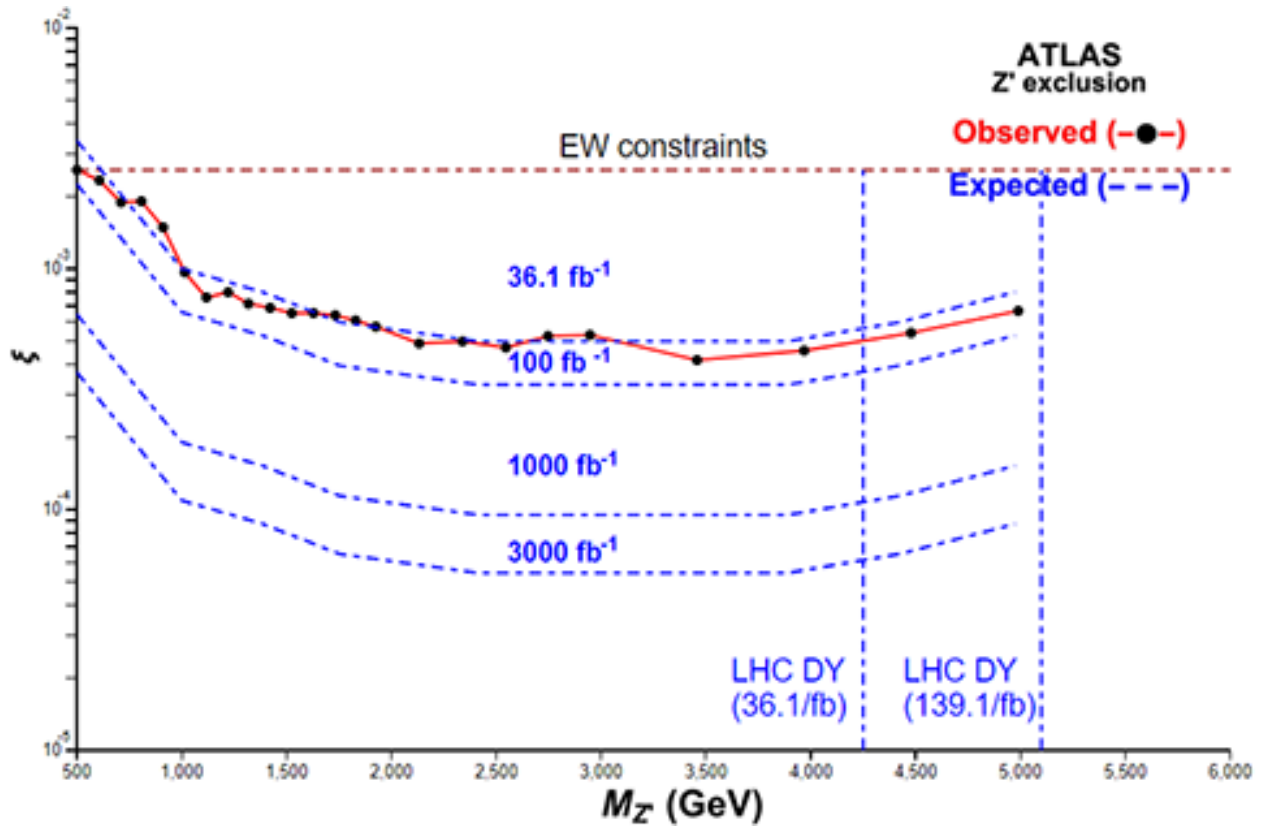


Рисунок 3.9 — Ограничения на угол смешивания $Z-Z'$ полученные из обработки данных имитационного моделирования

Исходя из того, что большая часть использованной литературы при исследовании проблемы описанной в данной диссертационной работе является англоязычной было принято решение реализовать возможность смены языка на котором отображается текст приложения. Такой функционал добавлен в правом верхнем углу *web*-интерфейса, рисунок 3.10.

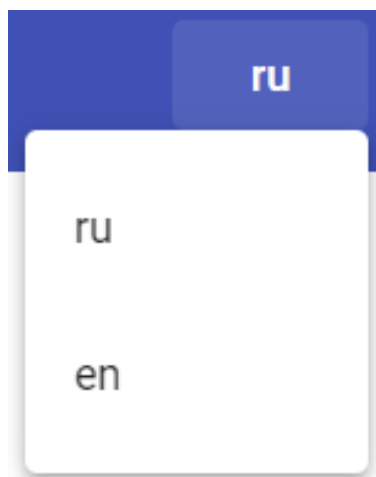


Рисунок 3.10 — Смена языка *web*-страницы

ГЛАВА 4

ТЕСТИРОВАНИЕ И ВЕРИФИКАЦИЯ РАЗРАБОТАННОГО ПРИЛОЖЕНИЯ

4.1 Верификация работы программы

Для верификации вычислений приложения использовались статьи написанные по реальным экспериментальным данным полученных от коллабора-ции *ATLAS*. Верифициционными данными являются ограничения на эффекты Z - Z' смешивания на Большом Адронном колайдере.

После открытия Z' -бозона на БАК через процесс ДЯ, необходимо произвести некоторую диагностику связей и смешивания Z - Z' , чтобы идентифицировать основную теоретическую структуру. В настоящей работе исследуются данные *ATLAS* [4] и *CMS* в канале дибозона.

$$pp \rightarrow W^+W^- + X \quad (4.1)$$

Для поиска Z -бозона, который возникает, например, в популярной модели с расширенным калибровочным сектором. Анализ основан на данных о столкновениях pp при энергии центра масс $\sqrt{s} = 13$ собранных группами *ATLAS* [4] и *CMS* на БАК. В частности, данные используются для поиска Z - Z' смешивание. На *ATLAS* события W^+W^- реконструируются через их полуплеитонные распады W , где один W -бозон распадается на заряженный лептон ($l = e, \mu$) и нейтрино, а другой на две струи, тогда как на *CMS* W -бозон адронически распадается на две восстановленные струи.

Процесс рождения пары W^-W^+ -бозонов 4.1 важен для изучения электрослабой калибровочной симметрии. Общие свойства слабых калибровочных бозонов тесно связаны с нарушением электрослабой симметрии и структуры калибровочного сектора, как и существование и структура трилинейных связей. Кроме того, канал распада дибозонов Z' исследует толщину калибровочной связи между новым и калибровочными бозонами стандартной модели. Кроме того, сила связи очень влияет на элементы распада и естественную ширину такого нового калибровочного бозона. Таким образом, детальное рассмотрение процесса 4.1 с высокой точностью проверяет калибровочный сектор СМ и может пролить свет на бозоны, которые могут появиться за пределами СМ. Здесь мы рассмотрим возможность наблюдения Z' -бозона в W^+W^- парного процесса на БАК, который в отличие от процесса ДЯ не является основным каналом поиска, но может помочь

понять происхождение новых калибровочных бозонов.

Для верификации верхних пределов сечения распада $Z' \rightarrow W^+W^-$ использована научная статья основанная на реальных экспериментальных данных [16]. На рисунке 4.1 изображены экспериментальные распределения по теориям, что полностью соответствует распределению построенного разработанным приложением (рисунок 3.8).

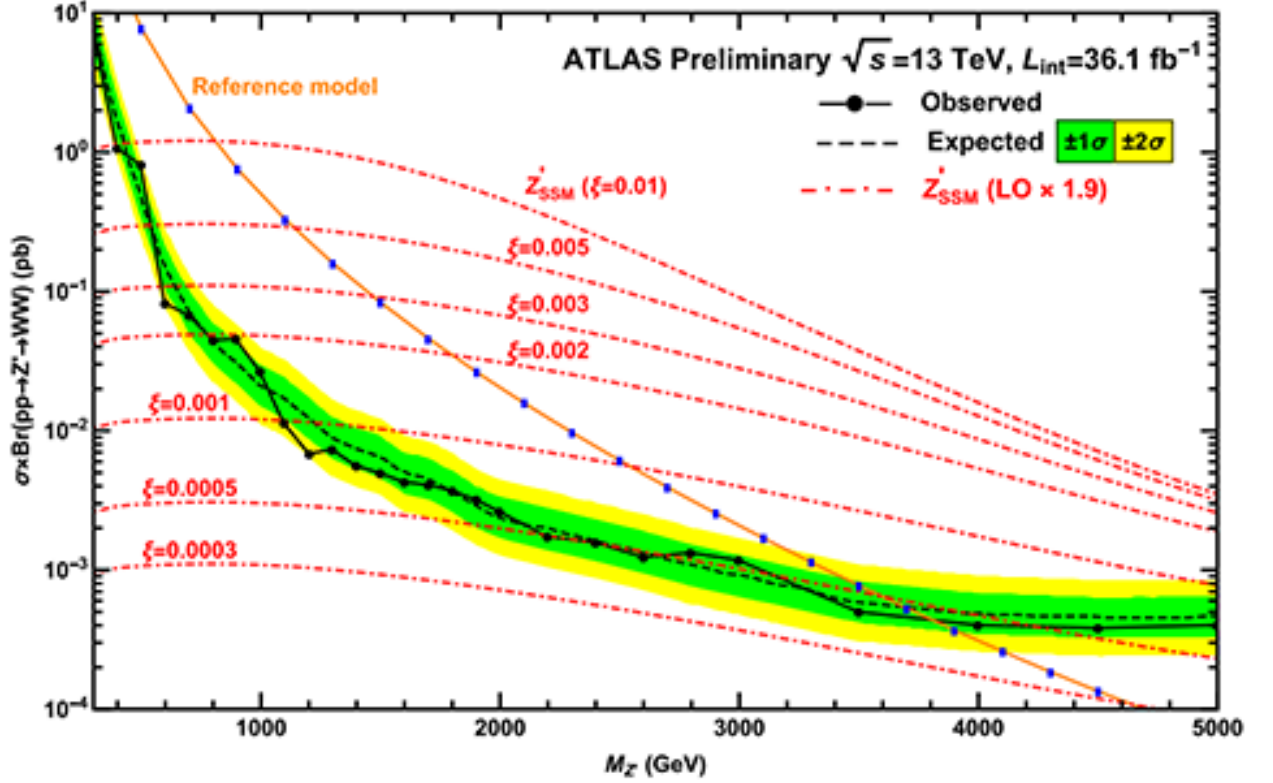


Рисунок 4.1 — Наблюдаемые и ожидаемые верхние пределы сечения распада $Z' \rightarrow W^+W^-$, как функция Z' массы $M_{Z'}$, полученные из обработки данных эксперимента *ATLAS*

Поиски тяжелого WW резонанса были выполнены на Теватроне исследовательскими группами *CDF* и *D0*. Группа *D0* изучала резонансное рождение дибозонов до 700 ГэВ в каналах распада $lv'l'v'$ и $lvjj$ [5]. Группа *CDF* также исследовала резонанс в WW в канале распада $evjj$, что в результате привело к обнаружения нижних лимитов масс Z' и W' -бозонов, за исключением масс превышающих 900 ГэВ, зависящих от параметра смешивания.

Исследования WW -резонансов группами *ATLAS* и *CMS* с использованием, соответственно, полулептонных и адронных событий распада в pp столкновениях при 13 ТэВ устанавливают массовые пределы 3 ТэВ для этих резонансов [4].

В этой работе изучается возможность рождения нового резонанса нейтрального спина 1 (Z') из доступных данных групп *ATLAS* и *CMS* для W^+W^- распадов. В качестве результатов работы будут получены ограни-

чения на соответствующие $Z - Z'$ -коэффициенты смешивания и на массу $M_{Z'}$.

Существует много теоретических моделей, которые предсказывают Z' с массой в диапазоне ТэВ. Мы рассмотрим модель, где Z' взаимодействуют со светлыми кварками и заряженными калибровочными бозонами посредством их смешивания с Z , предполагая, что Z -связывания имеют ту же структуру Лоренца, как и в СМ. В частности, в настоящем анализе мы сосредоточимся на калибровочном бозоне «последовательной стандартной модели» (ПСМ).

В простой модели связи Z бозонов с фермионам (кварки, лептоны) и W -бозонами являются прямой транскрипцией соответствующих связей стандартной модели. Заметим, что такого бозона Z' не ожидается в контексте калибровочных теорий, если он не имеет дополнительных связей с экзотическими фермионами. Однако он служит полезным справочным примером при сравнении ограничений из разных источников. Он также мог бы играть роль возбужденного состояния обычного Z в моделях композитности или с дополнительными размерами в слабом масштабе.

Во многих расширенных калибровочных моделях, в то время как связи с фермионами мало чем отличаются от связей стандартной модели, $Z'WW$ существенно подавляется по отношению к СМ. Фактически, в расширенной калибровочной модели стандартная модель трилинейной калибровочной связи бозонов, $g_{WWZ}(= \cot \theta_W)$ заменяется на $g_{WWZ} \rightarrow \xi \cdot g_{WWZ}$, где $\xi = C \cdot (M_W/M_{Z'})^2$ коэффициент смешивания и C - коэффициент масштабирования прочности связи [5]. Мы установим границы сечения для таких Z'_{SSM} как функция массы $M_{Z'}$ и ξ . Следует отметить, что большинство результатов поиска Z' показывают, что массовые пределы $\xi = (M_W/M_{Z'})^2$.

Изучение процесса рождения W^+W^- бозонной пары на Большом адронном коллайдере позволяет исследовать спонтанное нарушение калибровочной симметрии стандартной модели (SM) и может использоваться для поиска новых явлений за пределами SM . Дополнительные нейтральные векторные Z' -бозоны, распадающиеся на заряженные калибровочные векторные пары бозонов W^+W^- , предсказаны во многих сценариях новой физики, включая модели с расширенным калибровочным сектором. Процесс $pp \rightarrow W^+W^-$ позволяет установить жесткие ограничения на угол смешивания ξ для $Z-Z'$ и массу Z' , $M_{Z'}$. В настоящей работе впервые получены ограничения на параметры Z' смешивания в плоскости $\xi-M_{Z'}$, полученные из экспериментальных данных *ATLAS* и *CMS* на *LHC* при энергии 13 ТэВ и светимостями 36,1 и 35,9 фб⁻¹, соответственно.

Многие новые физические сценарии (NP) за пределами SM , включая суперструнные и лево-симметричные модели, предсказывают существова-

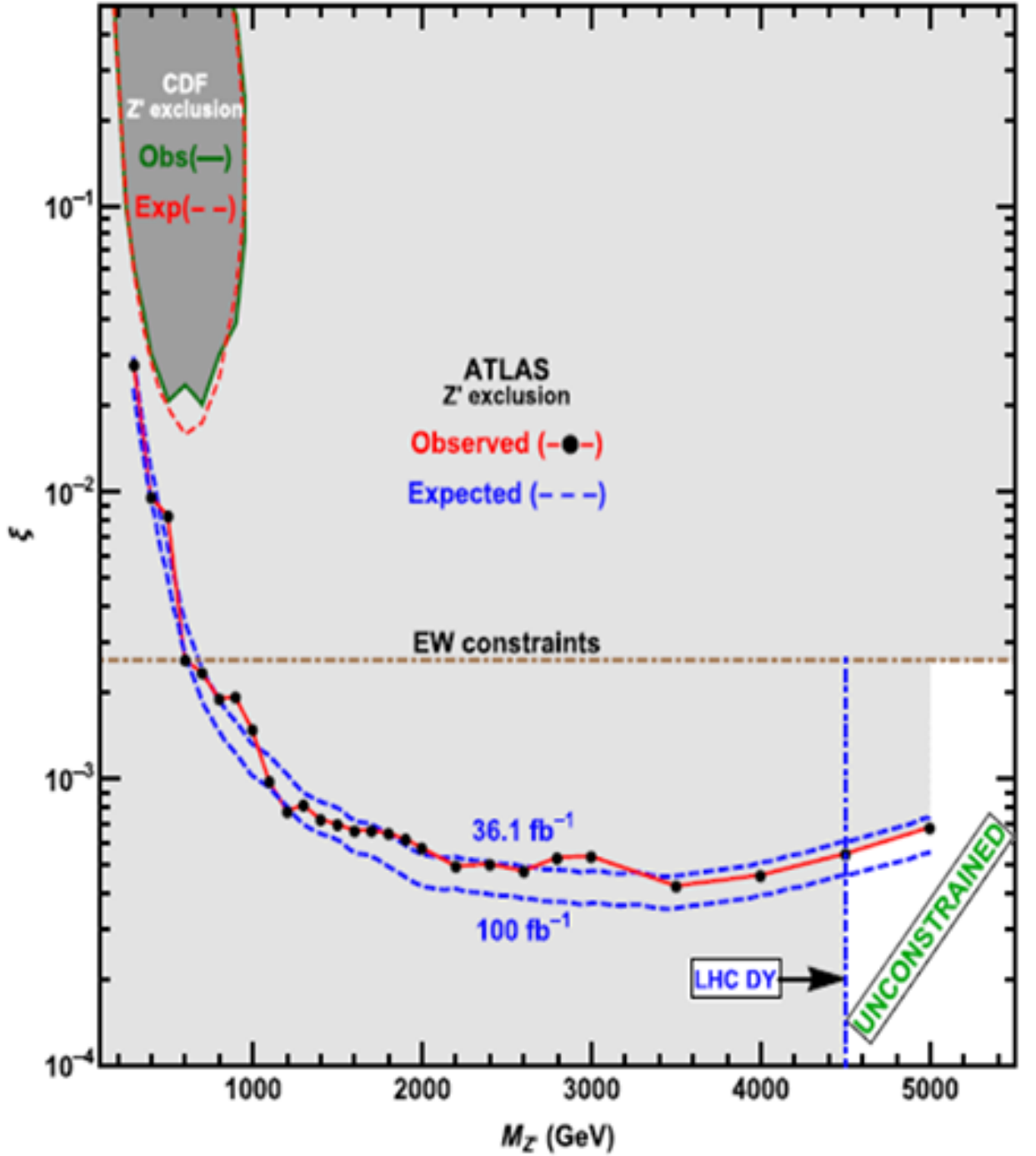


Рисунок 4.2 — Ограничения на угол смешивания Z - Z' полученные из обработки данных эксперимента *ATLAS*

ние новых нейтральных и заряженных калибровочных бозонов, которые могут быть достаточно легкими, чтобы быть доступными на текущих и / или будущих коллайдерах. Поиск этих новых нейтральных Z' и заряженных W' калибровочных бозонов является важным аспектом экспериментальной программы физики высокоэнергетических коллайдеров. Здесь рассматриваются эффекты Z' -бозонов. Существующие ограничения на прямое рождение на *LHC* и виртуальные эффекты на Большом электронно-

позитронном коллайдере (LEP) путем интерференции или смешения с Z -бозоном подразумевают, что любой новый Z' -бозон довольно тяжелый и очень мало смешивается с Z -бозоном. В зависимости от рассматриваемой теоретической модели массы Z' порядка 4,5 ТэВ и углы смешивания Z - Z' на уровне нескольких промилле исключены. Угол смешивания сильно ограничен очень высокоточными экспериментами на LEP и линейным коллайдером $SLAC$ (SLC).

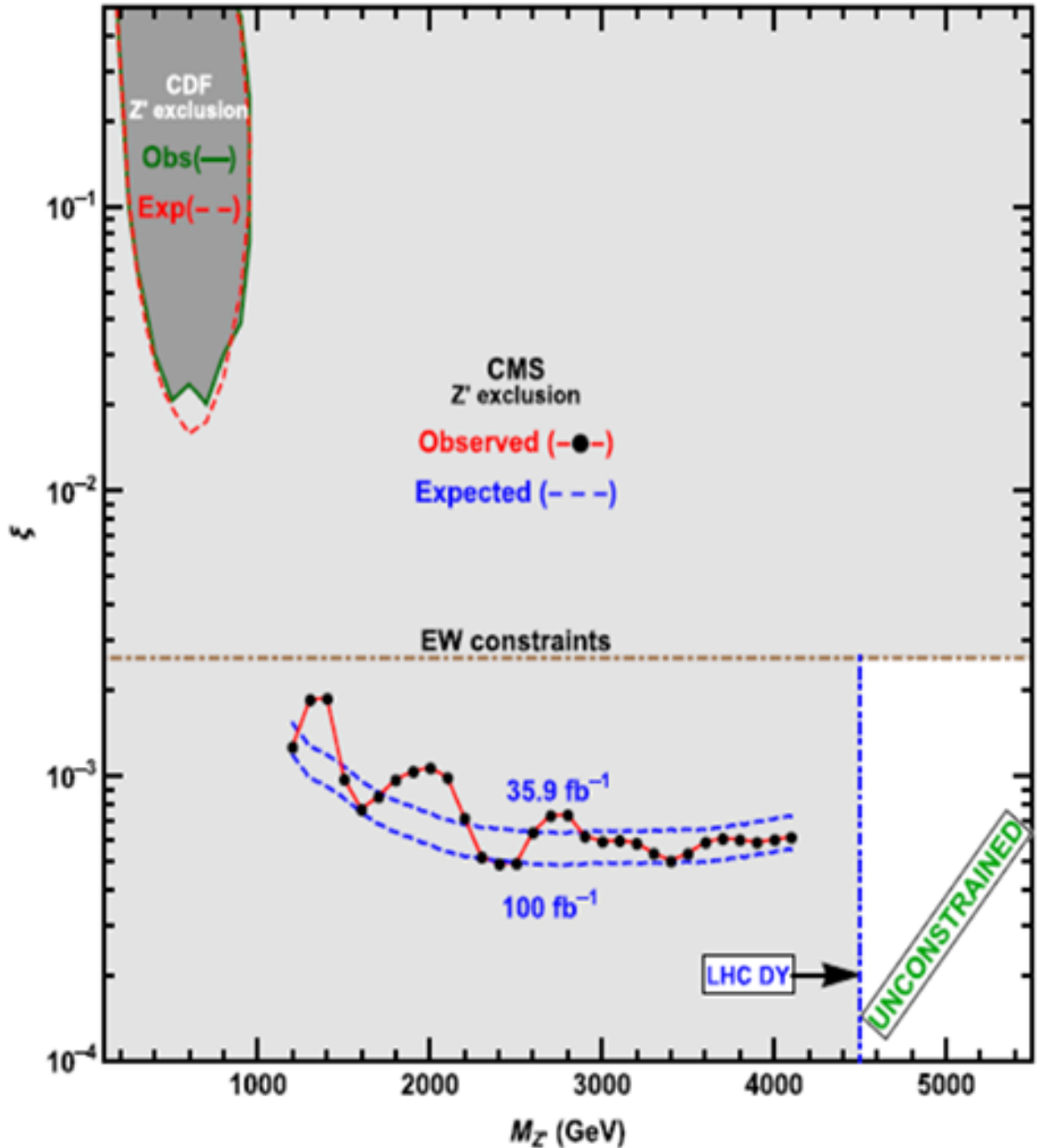


Рисунок 4.3 — Ограничения на угол смешивания Z - Z' полученные из обработки данных эксперимента CMS

Более подробно детали анализа данных экспериментов *ATLAS* и *CMS* представлены в работе [16].

Основной задачей приложения является получение ограничения на угол смешивания Z' -бозонов в протон-протонных столкновениях. На рисунке 3.9 изображены ограничения построенные по данным имментационного моделирования, которые и соответую полученным экперементальным данным $\xi < 0,0004$ (рис. 4.2, рис. 4.3), а так же рассчитаны ограничения для светимостей 1000 фб^{-1} и 3000 фб^{-1} , которые составили 10^{-4} и $6 * 10^{-5}$ соответственно.

4.2 Анализ результатов верификации

Во время выполнения приложения могут возникать различные исключительные ситуации, которые могут пагубно сказаться на процессе работы приложения, поэтому эти ошибки необходимо предусмотреть и сделать их обработку.

Для возможности ввода данных в приложение используется обычные веб элементы ввода данных пример которого можно увидеть на рисунке 4.4.



Рисунок 4.4 — Стандартный блочный элемент ввода данных

Разработанное приложение имеет встроенные валидаторы для того, чтобы пользователь не смог ввести данные, непредназначенные для конкретного поля. Например, если при обработке численных значений (присваивании значений в поля численных данных, выполнении математических операций) в незащищённое приложение ввести текст, то при попытке преобразований пользователь получит сообщение о критической ошибке, как показано на рисунке 4.5.



Рисунок 4.5 — Уведомление об неверно введенных данных

Благодаря инструмента валидации включенным в фреймворк *Angular 7*

можно легко валидировать *HTML* элементы в форме отправки запросов, что максимально исключает вероятность экстренного прекращения работы разработанного приложения. На рисунке 4.6 показан код на языке *TypeScript*, который выполняет проверку полей формы.

```
requestData = this.fb.group({
  ksi: [null, [Validators.required, DoubleValidator.isDouble]],
  mass: [null, [Validators.required, DoubleValidator.isDouble]],
  events: [null, [Validators.required]],
  cycles: [null, [Validators.required]]
});
```

Рисунок 4.6 — Код валидации формы

Моделирование столкновений протонов занимает много процессорного времени и для того, чтобы снизить загрузженность вычислительной системы имитационным моделированием было решено сделать верификацию и кэширование уже рассчитанных результатов.

Данные в кэше хранятся на устройстве с быстрым доступом и могут использоваться совместно с программными компонентами. Основная функция кэша – ускорение процесса извлечения данных. Он избавляет от необходимости проводить имитационное моделирование вновь и вновь для одинаковых начальных условий.

Данная задача легко решается по средствам стандартных инструментов кэширования фреймворка *Spring*, как показано на рисунке 4.7 необходимо лишь подключить аннотацию фреймворка.

```
@Cacheable(value = "cache", key = "#{ksi, #mass, #countOfPythiaEvents, #countOfRecalculate}")
@Override
public String getResult(final String ksi, final String mass, final String countOfPythiaEvents
```

Рисунок 4.7 — Код кэширования результата имитационного моделирования

Сохранение результата метода *getResult* для сервиса *ZprimeServiceImpl* достаточно, чтобы все результаты вычислений были сохранены. Так же необходимо указать базу данных для хранения кэш данных. В разработанном приложении была использована база данных *Redis*.

Redis (*REmote DIctionary Server*) – это нереляционная высокопроизводительная СУБД. *Redis* хранит все данные в памяти, доступ к данным осуществляется по ключу. Опционально копия данных может храниться на диске. Этот подход обеспечивает производительность, в десятки раз превосходящую производительность реляционных СУБД, а также упрощает

секционирование (шардинг) данных.

Важным моментом в реализации архитектуры разработанного приложения является возможность экспорта сохраненных данных из приложения (*persistence*). *Persistence Data* – это данные, сохраняющиеся в информационной системе в течение более одного сеанса управления данными. Компания Amazon предлагает широкий набор различных готовых решений в том числе и решение для сохранения данных между сеансами разработанного приложения называемое *AWS ElastiCache*. Данный сервис позволяет легко экспортировать данные из базы *Redis* в *Amazon S3 bucket*, как показано на рисунке 4.8.

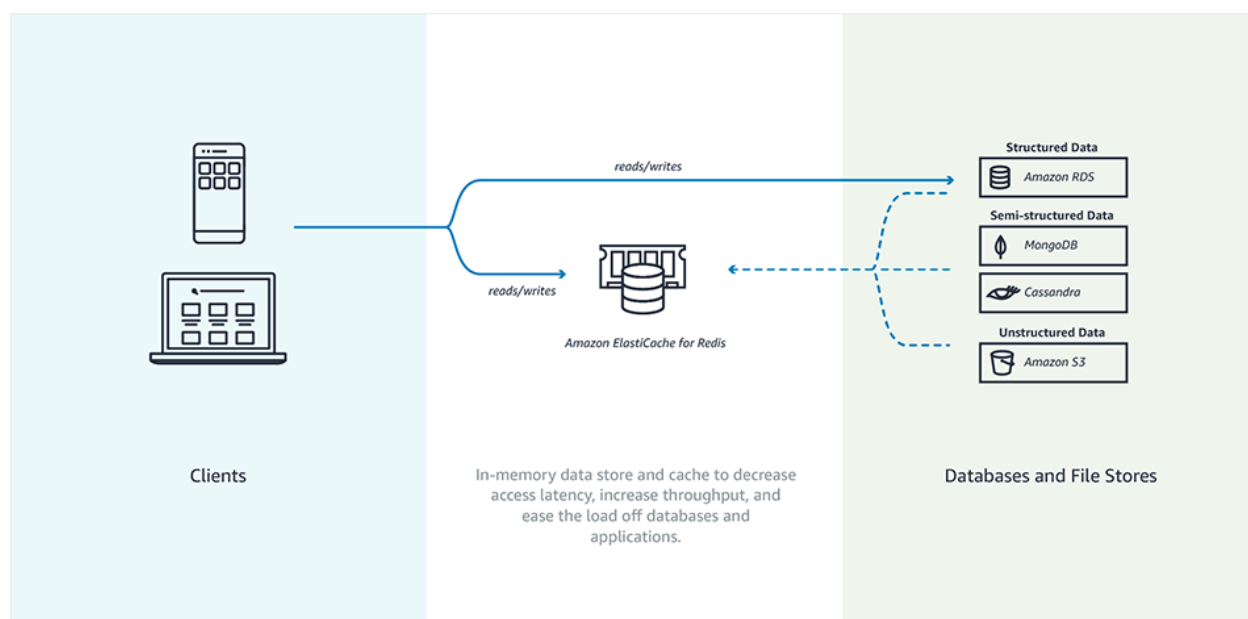


Рисунок 4.8 – Работа с данными в *AWS ElastiCache*

Amazon S3 (*Amazon Simple Storage Service*) позволяет хранить контент и получать к нему доступ из любого места в любое время. *Amazon S3* хранит данные в виде объектов в корзинах (*bucket*). Каждый объект представляет собой файл и, опционально — метаданные, которые этот объект описывают. Корзина — это контейнер, который хранит объекты. Можно иметь неограниченное количество таких корзин, и для каждой — установить доступ (кто может создавать, удалять и просматривать список объектов в корзине), просматривать логи доступа к корзине и объектам в ней.

ЗАКЛЮЧЕНИЕ

Выполнена обработка экспериментальных данных коллаборации *ATLAS* на Большом адронном коллайдере *LHC* (с энергией 13 ТэВ и светимостью $36,1 \text{ фб}^{-1}$) по измерению процесса рождения W^+W^- пар в протон-протонных столкновениях.

Была разработана математическая модель и имментационная модель, проведено имметационное моделирование и по результатам моделирования были получены ограничения на угол смешивания Z' -бозонов в модели *SSM*, которые составили $\xi < 0,0004$, что на порядок лучше результатов полученных ранее из глобального анализа электрослабых данных. А так же рассчитаны ограничения для светимостей 1000 фб^{-1} и 3000 фб^{-1} , которые составили 10^{-4} и 6×10^{-5} , соответственно.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Леонтьев, В. В. Информационные методы в физике высоких энергий / В. В. Леонтьев, И. И. Белотелов — Москва : «Университетская книга», 2011.
2. Alioli, S. NLO and parton showers: the POWHEG-BOX / Simone Alioli // DESY, Platanenallee 6, 15738 Zeuthen, Germany – 2013. – Vol. 40. – P. 12.
3. Event generation with SHERPA / Gleisberg, T. [et al.] // Phys. Rev. D. – 2014. – Vol. 60. – P. 3.
4. Java: the complete reference. / Schildt H. // McGraw-Hill Education Group, 2016. – P. 820.
5. Aspectj in action: enterprise AOP with spring applications / Laddad R. // Manning Publications Co., 2014. – P. 254.
6. Docker in action / Nickoloff J. // Manning Publications Co., 2016. – P. 304.
7. Amazon web services in action / Wittig A. // Manning Publications Co., 2015. – P. 500.
8. За пределами Стандартной модели [Электронный ресурс]. — Режим доступа: <https://elementy.ru/LHC/HEP/SM/beyondSM> — Дата доступа: 11.12.2017.
9. Designing the User Interface: Strategies for Effective Human-Computer Interaction / B. Shneiderman [et al]. — Pearson Education Limited, 2017. — 624 p.
10. Н. В. Красников, В. А. Матвеев. Поиск новой физики на LHC [Электронный ресурс]. — Режим доступа: http://nuclphys.sinp.msu.ru/ATLAS_exp/at03.htm — Дата доступа: 11.12.2017.
11. Official documentation [Электронный ресурс]. — Режим доступа: <http://home.thep.lu.se/torbjorn/Pythia.html> — Дата доступа: 11.12.2017.

12. Бобовников, И.Д. Эффекты $Z - Z'$ -смешивания в процессах рождения пары W^\pm -бозонов на адронных и лептонных коллайдерах высоких энергий / И.Д. Бобовников, А.А. Панков. — Письма в ЭЧАЯ, 2016. Т. 13, №1(199). С.8-35
13. Слабое взаимодействие [Электронный ресурс]. — Режим доступа: <http://nuclphys.sinp.msu.ru/enc/e149.htm> — Дата доступа: 11.12.2017.
14. Красников, Н.В. Новая физика на Большом адронном коллайдере / Н.В. Красников, В.А. Матвеев // М.: КРАСАНД, 2011. — 208 с.
15. Емельянов, В.М. Стандартная модель и ее расширения // М.: ФИЗМАТЛИТ. — 2007 — №584 — С. 55
16. Osland, P. Probing $Z - Z'$ mixing with ATLAS and CMS resonant diboson production data at the LHC at $\sqrt{s} = 13$ TeV / P. Osland, A.A. Pankov, A.V. Tsytrinov // Physical Review D. — 2017. — Vol. 96. — P. 055040.
17. Andreev, V. V. Constraints on the $Z - Z'$ mixing angle from data measured for the process $e^+e^- \rightarrow W^+W^-$ at the LEP2 collider / V.V. Andreev, A.A. Pankov // Phys. At. Nucl. — 2012. — Vol. 75. — P. 76.
18. Patrignani, C. Review of Particle Physics / C. Patrignani et al. [Particle Data Group] // Chin. Phys. C. — 2016. — Vol. 40, No.10. — P. 100001.
19. Search for massive resonances decaying into WW , WZ or ZZ bosons in proton-proton collisions at $\sqrt{s} = 13$ TeV / Sirunyan, A. M. [et al.] // J. High Energy Phys. — 2017. — Vol. 162. — P. 56.
20. Measurement of W^+W^- -production in pp collisions at $\sqrt{s} = 7$ TeV with the ATLAS detector and limits on anomalous WWZ and WW_γ couplings / Ada, G. [et al.] // Phys. Rev. D. — 2013. — Vol. 88. — P. 29.
21. Search for new phenomena in the $WW \rightarrow l\nu l'\nu'$ final state in pp collisions at $\sqrt{s} = 7$ TeV with the ATLAS detector / Ada, G. [et al.] // Physics Letters B. — 2013. — Vol. 3. — P. 878.

ПРИЛОЖЕНИЕ А

main-graph.service.ts

```
import {Injectable} from '@angular/core';
import {MainGraph} from '../graphD3/main-graph';
import {WebsocketService} from
    "../service/websocket.service";

@Injectable({
  providedIn: 'root'
})
export class MainGraphService {

  private MAIN_GRAPH_SELECTOR: string = 'div#svg';
  private LOCAL_STORAGE_PREFERENCES_KEY: string =
    'preferences';
  private mainGraphInstance: MainGraph;

  constructor(private websocketService:
    WebsocketService) {

  }

  ngOnDestroy() {
    this.mainGraphInstance.onGraphDestroy();
  }

  public renderMain() {
    this.mainGraphInstance = new
      MainGraph(this.MAIN_GRAPH_SELECTOR,
        this.websocketService);
    this.mainGraphInstance.init();
    // render preference
    this.requestPreferenceLines();
  }

  public requestPreferenceLines() {
```

```

let preferences =
    JSON.parse(localStorage.getItem(this.LOCAL_STORAGE_PREFE
if (!preferences) {
return;
}
preferences.forEach(lineKey => {
let lineInfo = lineKey.split('_');
this.mainGraphInstance.requestWholeLine(lineInfo[0],
    lineInfo[1], lineInfo[2]);
});
}

public getLinesInfo(): Array<string> {
return this.mainGraphInstance.getLinesKsi();
}

public requestLine(ksi, events, cycles) {
let lineKey = ksi + '_' + events + '_' + cycles;
if (!this.getLinesInfo().includes(lineKey)) {
this.addPreference(lineKey);
this.mainGraphInstance.requestWholeLine(ksi, events,
    cycles);
}
}

public addPreference(lineKey) {
let preferences =
    JSON.parse(localStorage.getItem(this.LOCAL_STORAGE_PREFE
if (!preferences) {
preferences = [];
}
preferences.push(lineKey);
localStorage.setItem(this.LOCAL_STORAGE_PREFERENCES_KEY,
    JSON.stringify(preferences));
}

public removePreference(lineKey) {
let preferences =
    JSON.parse(localStorage.getItem(this.LOCAL_STORAGE_PREFE
let index = preferences.indexOf(lineKey);

```

```
preferences.splice(index, 1);
localStorage.setItem(this.LOCAL_STORAGE_PREFERENCES_KEY,
    JSON.stringify(preferences));
}

public removeLine(lineKey) {
    console.log('Remove: ' + lineKey);
    this.removePreference(lineKey);
    this.mainGraphInstance.removeLine(lineKey);
}

}
```