

ПРАВИТЕЛЬСТВО РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ  
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ  
«САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»  
(СПбГУ)

Кафедра физики высоких энергий и элементарных частиц  
Направление «Физика»



**Влияние эффекта слияния цветных кварк-глюонных струн на  $FB$   
корреляции множественности в столкновениях  
ультрарелятивистских ядер при высоких энергиях**

Курсовая работа

**Ужвы Дениса Романовича**

Научный руководитель:  
д.ф.-м.н., проф. **Вечернин В.В.**

Санкт-Петербург  
2018

# Содержание

<b>1</b>	<b>Введение</b>	<b>3</b>
1.1	Модельный статистический анализ в ФВЭиЭЧ . . . . .	3
1.2	Модель кварк-глюонных струн в КХД . . . . .	3
1.3	Слияние струн . . . . .	4
1.4	Генераторы событий Монте-Карло в мягкой области . . . . .	4
1.5	Постановка задачи . . . . .	5
<b>2</b>	<b>Разработка генератора событий</b>	<b>5</b>
2.1	План разработки и расчётов . . . . .	5
2.2	Выбор среды разработки . . . . .	5
2.3	Структура программы . . . . .	6
2.3.1	Генерация струн . . . . .	6
2.3.2	Заполнение графа и нахождение компонент связности; поиск максимальной компоненты . . . . .	7
2.3.3	Количество струн в максимальном кластере и его площадь . . . . .	7
2.3.4	Генерация множественности вперёд-назад . . . . .	7
2.4	Внутрипрограммный расчёт корреляционного коэффициента . . . . .	8
<b>3</b>	<b>Результаты расчётов</b>	<b>8</b>
3.1	Расчёты в MATLAB . . . . .	8
3.2	Расчёты с помощью программы на C++ . . . . .	9
3.2.1	Перколяционный переход . . . . .	9
3.2.2	Учёт слияния струн до адронизации . . . . .	9
3.2.3	Вычисление корреляционного коэффициента . . . . .	10
<b>4</b>	<b>Заключение</b>	<b>10</b>
<b>5</b>	<b>Список литературы</b>	<b>11</b>
<b>6</b>	<b>Приложение</b>	<b>13</b>
6.1	Изображения . . . . .	13
6.2	Таблицы . . . . .	19
6.3	Код программы . . . . .	21

# 1 Введение

## 1.1 Модельный статистический анализ в ФВЭиЭЧ

В современной физике высоких энергий и элементарных частиц верификация феноменологической гипотезы не всегда возможна прямым сравнением данных из эксперимента с предсказанием гипотезы вследствие вероятностного характера описываемых величин. Данная характеристика порой влечёт за собой чрезмерную сложность формульного анализа, необходимого для конкретных математических результатов. Для проверки такого рода гипотез на помощь приходят компьютерные симуляции, позволяющие эффективно набрать необходимую выборку статистических данных для анализа.

Симуляции представляют из себя, во-первых, модельное воспроизведение так называемых “событий” (event generation) – физических процессов, необходимых для наблюдения определённых явлений, в виде сгенерированного на основе гипотезы набора величин; во-вторых, машинные расчёты по генерируемым значениям величин – для удобного сравнения результатов симуляций с аналогичными результатами на основе экспериментальных данных [1]. К типичным моделируемым в рамках физики высоких энергий процессам можно отнести, к примеру, процесс множественного рождения частиц при столкновении (рассеянии) ядер в коллайдере.

На этапе генерации физических процессов, происходящих в коллайдере, можно провести прямую аналогию непосредственно с самой методикой эксперимента: для изучения какого-либо явления в физике элементарных частиц зачастую требуется большая выборка для уменьшения статистической погрешности. Так, например, после автоматической селекции “интересных” событий в ЛНС скорость их счёта составляет порядка сотен в секунду (до селекции – порядка сотен миллионов в секунду), что соответствует передаче информации на скорости приблизительно 1 ГБ/с. Дальнейший анализ оставляет на обработку намного меньше интересующих событий: за период работы с 2011 по 18 июня 2012 года количество собранных данных с детектора CMS позволило обнаружить лишь около 800-900 взвешенных по отношению сигнал-шум событий с инвариантной массой  $m_{\gamma\gamma} \approx 125$  GeV, соответствующих бозону Хиггса [2], но при этом погрешность была достаточно малой для утверждения об обнаружении ранее не наблюдавшейся частицы.

## 1.2 Модель кварк-глюонных струн в КХД

Квантовая хромодинамика (КХД) возникла в результате развития кварковой модели [3], и на данный момент является общепризнанной теорией сильных взаимодействий. Идея КХД заключается в присвоении кваркам так называемого цветового заряда, отвечающего за сильное взаимодействие подобно электрическому – за электромагнитное. Всего цветовых зарядов три (плюс три антицвета): красный, зелёный и синий (и парные антикрасный, антизелёный и антисиний соответственно). Характерная особенность сильного взаимодействия заключается в конфайнменте – невозможности существования кварков по отдельности: данные частицы могут быть лишь в “обесцвеченной” (“белой”) совокупности (обесцвечивание происходит по закону аддитивного смешения цветов, например, красного с зелёным и синими или зелёного с антизелёным). Причиной конфайнмента служит возрастание энергии поля сильного взаимодействия при отдалении кварков друг от друга, в отличие от того же электромагнитного поля. Стоит отметить также, что “цвета” кварков – лишь удобная репрезентация группы  $SU(3)$ , классифицирующей адроны, и не имеют отношения к оптическим цветам.

Столкновения ядер в коллайдере можно разделить на две категории: мягкие и жёсткие. При жёстком рассеянии частицы встречных ядер сталкиваются “лоб в лоб”, порождая так

называемые адронные струи с большим поперечным импульсом  $p_t$  [4]. До недавних пор физики принимали в расчёт только эту категорию столкновений, воспринимая мягкое рассеяние как возмущение. Важно также отметить, что расчёты в рамках жёсткой области можно проводить из первичных принципов, используя модели столкновений частиц как шаров. Позже было выяснено, что рассеяние в мягкой области нельзя свести к возмущению, так как на малых масштабах нарушаются сильные связи между партонами [5, 6].

С точки зрения феноменологии удобной описательной моделью мягкой адронизации является модель цветных кварк-глюонных струн [7–10]. Суть модели заключается в понимании кварк-глюонного поля в партонном облаке как совокупности струн, связывающих эти партоны. В момент столкновения двух ядер струны из встречных партонных облаков переплетаются, натягиваются и рвутся (фрагментируют); “обрывки” струн при этом вследствие конфайнмента рекомбинируют в адроны (см. рис. 1 в приложении).

### 1.3 Слияние струн

Множественность рождённых в столкновении адронов напрямую зависит от энергии налетающих частиц и от центральности столкновения. Пусть все рассматриваемые столкновения центральны, тогда чем больше энергия, тем больше струн – следовательно, процесс фрагментации более интенсивный. Это суть следствие из связи количества струн с количеством морских кварков [11]. Однако, при ядро-ядерных столкновениях на энергиях порядка ЛНС плотность струн может оказаться столь высокой, что приходится прибегать к нелинейным поправкам, вызванным эффектом взаимодействия струн между собой [12].

Для случая рассеяния тяжелых ядер М.А. Брауном и К. Пахаресом была предложена модификация струнной модели, учитывающая процессы возможного слияния первичных струн до начала процесса их фрагментации [13–16]. Слившиеся струны образуют так называемые кластеры (совокупности слившихся струн, отдельные от других подобных совокупностей), а множественность адронов зависит от параметров того или иного кластера, причём чем больше плотность струн (больше энергия), тем слабее прирост плотности струн влияет на прирост значения средней множественности [17]: адронизация происходит от одной большой “струны” вместо суммы составляющих её. Стоит отметить, что учёт нелинейности можно определять как при локальном сложении полей (локальное слияние), так и при сложении по всему поперечному сечению каждого кластера (глобальное слияние) [17].

За счёт слияния струн могут меняться корреляции множественности ( $nn$ ) и поперечного импульса ( $p_t p_t$ ) адронов в переднем и заднем быстротных окнах, возникающие благодаря флуктуациям числа участников столкновения (в струнной модели – числа струн). Причём на корреляции  $p_t p_t$  учёт кластеризации струн оказывает большее влияние, нежели на  $nn$  [17, 18]. Связано это с тем, что усреднённая по событию величина  $\langle p_t \rangle$  является интенсивной, в отличие от экстенсивной  $\langle n \rangle$ , и поэтому практически не зависит от числа струн, принимая малые значения при отбрасывании эффекта слияния. Однако, если кластеризацию учесть, начнут проявляться ощутимые отклонения  $\langle p_t \rangle$  от нуля, влекущие за собой наличие  $p_t p_t$  корреляций.

### 1.4 Генераторы событий Монте-Карло в мягкой области

В качестве компьютерной симуляции физики мягкой области рассеяния используются “Монте-Карловские генераторы” [19, 20] – пакеты программ, использующие, к примеру, модель кварк-глюонных струн для воспроизведения процесса адронизации. Примером такой программы может послужить широко используемый в теоретическом анализе инструмент Pythia. Данная программа использует решёточные симуляции [21, 22]: моделируется поперечное сечение ядро-ядерного столкновения, в котором происходит наложение распределённых сечений участвующих в адронизации струн. По измерениям площадей этих сечений и плотности струн можно рассчитывать  $nn$ ,  $p_t p_t$  и  $p_t n$  корреляции в различных быстротных

## 1.5 Постановка задачи

При всей удобности и высокой скорости расчётов решёточных симуляций интересно также рассмотреть более детальный подход: вместо дискретной решётки использовать непрерывную область сечения, так как непрерывная область позволяет точнее смоделировать сечение, что влечёт за собой более качественную проверку теории. Именно такой подход был использован для определения и исследования  $nn$  и  $p_t p_t$  корреляций в данной работе.

Результатом данной работы является качественное и количественное сравнение  $nn$  и  $p_t p_t$  корреляций в переднем и заднем (FB – Forward, Backward) быстротных окнах с учётом слияния струн и без, определение влияния учёта эффекта слияния струн на множественность адронов, а также сравнение результата с выкладками из [23].

## 2 Разработка генератора событий

### 2.1 План разработки и расчётов

Для реализации генератора решено использовать объектно-ориентированный подход с целью обеспечить удобство дальнейших усовершенствований программы. В самой программе первоначально генерируется некоторое количество поперечных сечений струн в виде кругов фиксированного радиуса по заданному распределению (радиус струн соответствует параметрам распределения пропорционально реальному масштабу), затем производятся необходимые расчёты. Принципиальная схема работы программы представлена на рис. 2 в приложении. Также в приложении (рис. 3) представлена визуализация генерируемых струн.

Для расчёта корреляций множественности и поперечного импульса адронов в переднем и заднем быстротных окнах (всюду далее величина корреляционного коэффициента будет обозначаться как  $b$ ) необходимо сгенерировать соответственно значения множественности  $n$  по сгенерированным струнами и значения поперечного импульса  $p_t$  по струнной конфигурации и полученным значениям поперечного импульса. В качестве реализации учёта эффекта слияния струн  $n$  и  $p_t$  рассчитывается из плотности наложения (пересечения) струн друг на друга подсчётом количества струн в отдельно взятом кластере, а также подсчётом поперечной площади данного кластера (как наиболее физичное далее рассматриваться будет только глобальное слияние).

Таким образом, разработка генератора разбивается на четыре этапа в порядке возрастания сложности, причём каждый следующий этап нуждается в завершённости предыдущих:

1. Генерация струн.
2. Поиск кластеров.
  - 2.1. Подсчёт количества струн в каждом кластере.
  - 2.2. Вычисление площади каждого кластера.
3. Генерация значений множественности и поперечного импульса.
4. Расчёт FB корреляций множественности и поперечного импульса.

### 2.2 Выбор среды разработки

Первоначально в качестве языка разработки был выбран MATLAB как очень удобный инструмент для быстрой и качественной реализации вычислительных алгоритмов. С помощью MATLAB были преодолены первые два этапа разработки, пока не был выявлен существенный недостаток данного решения – слишком невысокая скорость расчётов по сравнению с

более низкоуровневыми языками программирования, такими как C++ (приблизительно в 15-25 раз медленнее). На последнем впоследствии было заострено внимание, и далее было принято решение остановиться на нём для дальнейшей разработки.

Скорость расчётов и оптимизация кода особенно важна в данной работе, так как предполагается расчёт большого количества симуляций. Так, в программе была задействована технология OpenMP, позволяющая создавать многопоточные приложения.

## 2.3 Структура программы

Программа состоит из класса “Simulation”, представляющего собой модель одной симуляции, и “оболочки”, которая необходимое количество раз повторяет объект-симуляцию с загружаемым в неё извне средним количеством струн  $\langle N \rangle$ . Само значение  $\langle N \rangle$  находится из значения параметра  $\eta$  (средняя плотность струн, обозначается всюду далее без угловых скобок, так как сама по себе не является параметром для распределений) через соотношение

$$\eta = \frac{\langle N \rangle \sigma_0}{S}, \quad (1)$$

где  $\sigma_0$  – площадь одной струны,  $\sigma_0 = \pi r_s^2$ , где  $r_s = 0.2 \div 0.3$  фм – радиус струны;  $S$  – общая площадь взаимодействия в поперечном сечении ( $S = \pi R^2$ , тогда  $R \approx 7.5$  фм – радиус ядра, т.е. области взаимодействия). Так, для энергий RHIC значение  $\eta \approx 3$ , а для LHC  $\eta \approx 11$  [24]. Выбор значений  $\sigma_0$  и  $S$  требует некоторых отдельных пояснений. Во-первых, программа “разбрасывает” по факту не сами струны, а их центры, и тогда если центр струны ляжет на границу области взаимодействия, то часть струны будет внутри области, а часть – снаружи. Тогда, в качестве  $S$  следует принимать значение  $S = \pi(R + r_s)^2$ , где  $R = 7.5$  фм и  $r_s = 0.225$  фм фиксируются (выбор значения 0.225 объясняется тем, что при масштабировании  $R$  до значения 1 значение  $r_s$  масштабируется до 0.03). Во-вторых, в некоторых промежуточных тестовых стадиях разработки в качестве  $S$  берётся суммарная площадь всех кластеров, так как в качестве функции плотности распределения координат струн выступает гауссиан.

Класс симуляции состоит из нескольких основных (см. листинг 1 в приложении) и вспомогательных (листинг 2 в приложении) функций-методов. Каждый из основных методов класса выполняется один раз за симуляцию в соответствии с порядком в листинге. Описание данных методов следует ниже по порядку.

### 2.3.1 Генерация струн

Чтобы сгенерировать струны, изначально следует задать их количество  $N$ . Для расчёта  $b$  (4 этап) решено генерировать  $N$  с помощью равномерного дискретного распределения с математическим ожиданием равным  $\langle N \rangle$  на промежутке от 0 до  $2N$  (в предыдущих этапах в качестве  $N$  бралось непосредственно значение  $\langle N \rangle$ ).

В качестве функций плотности распределения (PDF) координат струн первоначально бралось равномерное непрерывное распределение в круге, затем более физичное гауссово распределение. Для этапа расчёта корреляций используется отражённая по оси  $z$  трёхмерная парабола

$$z = -x^2 - y^2 + 1 \quad (2)$$

как хорошее приближение гауссиана в интересующей области.

Генерация координат струн происходит методом Неймана как универсальным методом генерации значений случайной величины по любой функции плотности распределения (в перспективе – использование в качестве PDF довольно сложный потенциал Вудса-Саксона, поэтому использование метода Неймана оправдано). Метод Неймана заключается в следующем: на бруске, полностью покрывающем носитель функции плотности распределения  $f(x)$ , равномерно разыгрывается число  $A$ ; затем равномерно разыгрывается число  $B$  от 0 до 1, и если  $B < f(A)$ , то  $A$  утверждается как величина, распределённая по  $f(x)$ .

### 2.3.2 Заполнение графа и нахождение компонент связности; поиск максимальной компоненты

Граф пересечений струн представляет из себя симметричную матрицу  $N \times N$ , заполненную номерами струн, которые пересекаются со струной с номером соответствующей строки или столбца, либо нулевыми элементами (в случае, если струна под номером столбца или строки не пересекается ни с какой другой). Факт пересечения двух струн равносильно тому, что расстояние между их центрами меньше либо равно двум радиусам одной струны  $2r_s$  (проверяется с помощью 5-й вспомогательной функции из листинга 2 в приложении).

Для определения кластеров ищутся компоненты связности графа. Здесь используется метод Depth-first search: в качестве первой компоненты связности берётся первая струна  $s_1$ , затем производится поиск пересечённых с ней струн  $\{s'_i\}$  и берётся первая пересечённая струна  $s'_1$ , далее ищутся струны  $\{s''_i\}$ , пересечённые с  $s'_1$ , и так далее, пока пересечённых струн не останется (до  $s_1^{(n)}$ ); эта же процедура проводится для  $s_2^{(n-1)}$  и всех остальных из  $\{s_i^{(n-1)}\}$ , из  $\{s_i^{(n-2)}\}$  и так далее до самой  $s'_2$ . Одновременно “проверенные” струны вычёркиваются из списка поиска, и следующая компонента связности ищется по струне  $s_j$ , где  $j > 1$  – первый номер струны, не вошедшей в первую компоненту. Проще говоря, каждую компоненту связности можно представить как куст, состоящий из веток, веток этих веток и т.д. до самых листьев, а метод Depth-first search – счётчик узлов веток (листья тоже считаются узлом – на поверхности). Если этот счётчик установить в каком-нибудь узле внутри куста, то он будет на каждом шаге стремиться к листьям, чтобы посчитать сначала узлы на поверхности, затем уйдёт на первый слой узлов, предшествующий листьям, потом на второй и далее до самого глубокого слоя, где остановится. Слово “Depth” в названии метода произошло из терминологии в теории графов: глубокими уровнями компоненты называются именно поверхностные уровни куста. Наглядная схема начала обхода небольшого графа представлена на рис. 4 в приложении. Помимо неё в приложении представлен код, реализующий алгоритм поиска компонент методом DFS (листинг 3)

Максимальная компонента связности находится из полученного набора компонент как самая длинная, т.е. имеющая больше всего струн. Отыскание максимальной компоненты связности означает нахождение максимального кластера – нужного для проверки корректности работы программы.

### 2.3.3 Количество струн в максимальном кластере и его площадь

Зная номер максимальной компоненты связности, нетрудно вычислить количество струн  $N_{cl}$  в соответствующем кластере: оно равно длине массива (вектора) компоненты, так как этот массив хранит номера всех струн из кластера.

Менее тривиальной задачей является вычисление площади кластера. Здесь на помощь приходит метод Монте-Карло вычисления площадей фигуры произвольной формы: фигурой в данном случае является граничный контур наложенных друг на друга кругов-струн. Суть метода в следующем: на прямоугольник или круг, покрывающий фигуру, разбрасывается сетка из точек, и тогда площадь фигуры примерно равна площади прямоугольника, помноженной на отношение количества попавших на фигуру точек к количеству не попавших.

### 2.3.4 Генерация множественности вперёд-назад

С помощью описанных выше методов сперва вычисляется площадь каждого кластера  $S_k$  и количество струн в них  $N_k$ . Следуя [17], средняя множественность  $\langle n \rangle_k$  для  $k$ -го кластера (глобальное слияние) в данном быстротном интервале вычисляется как

$$\langle n \rangle_k = \mu_0 \frac{S_k}{\sigma_0} \sqrt{l_k}, \quad l_k = \frac{N_k \sigma_0}{S_k}, \quad (3)$$

где  $\sigma_0$  – площадь одной струны,  $\mu_0$  – средняя множественность от одной струны в данном быстройном интервале. Для единичного интервала быстройности  $\mu_0 = 1.1$  в соответствии с [25] (в дальнейших расчётах бралось  $\mu_0 = 1$ ). Формулу 3 можно упростить до

$$\langle n \rangle_k = \mu_0 \sqrt{\frac{N_k S_k}{\sigma_0}}, \quad (4)$$

подставив явно выражение для  $l_k$  в выражение для  $\langle n \rangle_k$ .

Найденное значение  $\langle n \rangle_k$  показывает лишь среднее число частиц, рождённых кластером  $k$ . Пусть в симуляции  $i$  кластер  $k$  имеет среднее значение множественности  $\langle n \rangle_{ki}$ . Для генерации значения множественности в переднем  $n_{ki}^F$  и заднем  $n_{ki}^B$  быстройных окнах используется распределение Пуассона с математическим ожиданием равным  $\langle n \rangle_{ki}$ . Таким образом, для каждого кластера получается своя выборка значений  $n_{ki}^F$  и  $n_{ki}^B$ , эти выборки суммируются для каждого события

$$\begin{aligned} n_i^F &= \sum_{k=1}^{M_i} n_{ki}^F, \\ n_i^B &= \sum_{k=1}^{M_i} n_{ki}^B, \end{aligned} \quad (5)$$

где  $n_i^F$  и  $n_i^B$  – FB множественности в событии  $i$ ,  $M_i$  – количество кластеров в событии  $i$ . Стоит отметить, что данные выкладки проводятся для фиксированного параметра  $\eta$ .

## 2.4 Внутрипрограммный расчёт корреляционного коэффициента

Выборки значений  $\{(n_F)_i\}$  и  $\{(n_B)_i\}$  непосредственно используются в расчёте  $b$  с помощью вспомогательного метода 1 (см. листинг 2 в приложении). Для фиксированного  $\eta$  в соответствии с [23] величина  $b$  находится как

$$b = \frac{\langle n_F n_B \rangle - \langle n_F \rangle \langle n_B \rangle}{\langle n_F^2 \rangle - \langle n_F \rangle^2}, \quad (6)$$

усреднение происходит по всем симуляциям. Данный расчёт в самой программе удобен лишь для оценки правильности её работы, потому как часто для увеличения выборки приходится использовать данные с симуляций, рассчитанные на нескольких машинах (распределённые вычисления на данный момент не реализованы).

# 3 Результаты расчётов

## 3.1 Расчёты в MATLAB

Как указано выше, первоначально алгоритм расчётов был реализован в среде MATLAB. Функции, которые были применены в разработке на этом языке, аналогичны описанным выше для C++ за исключением уже встроенной в MATLAB функции `conncomp(G)`, возвращающей компоненты связности графа  $G$ .

С помощью MATLAB первые два этапа работы были преодолены с положительным результатом. Данный результат заключается в наблюдении так называемого “перколяционного перехода”: фазового перехода кластеризации, при котором в некоторой области  $[\eta, \eta + \Delta\eta]$  резко возрастает размер максимального кластера при малом увеличении  $\eta$ . Под “размером” максимального кластера в зависимости от задачи подразумевается как количество струн в нём  $N_{cl}$ , так и его площадь  $S_{cl}$ . С целью наблюдения перколяционного прехода удобно построить графики зависимостей  $N_{cl}/N$  и  $S_{cl}/S$  от  $\eta$ , что и было сделано (рис. 5 и 6 в приложении)



для небольшого количества симуляций. На обоих графиках можно выделить три интервала: при  $\eta$  от 0 до  $0.2 \div 0.3$ ;  $\eta$  от  $0.2 \div 0.3$  до  $1.8 \div 1.9$ ;  $\eta$  от  $1.8 \div 1.9$  до 10.2. Первый интервал соответствует слабому росту размера максимального кластера при добавлении струн: межструнное расстояние довольно большое при столь малом  $\eta$  (соответствующем количеству струн  $N = 0 \div 500$ ), поэтому самые большие кластеры состоят из небольшого количества струн  $N_{cl}$  в сравнении с  $N$ , а при увеличении  $N$  в среднем количество струн в них увеличивается медленно – вместо этого происходит лишь рождение новых кластеров с  $N_{cl}$  примерно равным “максимальным”. Далее по мере роста  $N$  и приближения ко второму интервалу происходит уплотнение струн в области взаимодействия и воссоединение небольших кластеров, причём каждая новая струна может соединить один или несколько максимальных кластеров в один значительно больший кластер (например, три близких кластера из 20 струн каждый могут соединиться добавленной струной в кластер из 61 струны); и такая соединительная струна может попасться в любой симуляции при заданном  $N$ , что вызывает довольно большую погрешность значений  $N_{cl}/N$  и  $S_{cl}/S$  во втором интервале значений  $\eta$ . В третьем интервале происходит своеобразное насыщение: при больших  $N$  струны всё плотнее заполняют область взаимодействия, сливаясь в единственный большой кластер; как следствие,  $N_{cl} \rightarrow N$  и  $S_{cl} \rightarrow S$ .

Стоит отметить, что расчёт зависимости  $S_{cl}/S$  от  $\eta$  для 60 симуляций занимал около полутора-двух часов.

## 3.2 Расчёты с помощью программы на C++

### 3.2.1 Перколяционный переход

Расчёты, аналогичные проделанным в MATLAB, зависимости  $N_{cl}/N$  и  $S_{cl}/S$  от  $\eta$  проведены успешно, о чём свидетельствует ожидаемый вид соответствующих графиков (рис. 7 и 8 в приложении). Можно заметить, что область перколяционного перехода теперь не такая резкая, какой была при вычислениях в MATLAB. Данный результат связан с заменой равномерного распределения гауссианом: насыщение происходит медленнее вследствие уплотнения струн радиально к центру (это приводит к формированию отдельных небольших кластеров около периметра области взаимодействия, где плотность струн низкая). Наличие погрешностей  $\Delta\eta$  (см. таблицу 3 в приложении) связано с вычислением значения  $\eta$  для каждого события в связи с переменной площадью области взаимодействия из-за использования гауссового распределения.

Программа провела 3638 симуляций с вычислением как  $N_{cl}/N$ , так и  $S_{cl}/S$ , и на оба расчёта ушло 5 часов суммарно. Это значит, что расчёт 60 симуляций занял бы приблизительно 5 минут, что намного быстрее вычислений в MATLAB.

### 3.2.2 Учёт слияния струн до адронизации

Описанные выше результаты получены благодаря преодолению первых двух этапов разработки. Чтобы пронаблюдать явно учёт влияния кластеризации на множественность адронов в событии и проверить слова из [17], необходимо работать уже с третьим этапом. В вычислениях на третьем этапе также используется гауссиан.

Описанный выше процесс генерации множественности в программе используется для сравнения множественности адронов  $\langle n \rangle$ , рождённых каждым кластером в совокупности событий, с множественностью  $\langle n_0 \rangle$ , полученную от каждой струны по отдельности (усреднение данных величин ведётся по совокупности событий при фиксированной величине  $\eta$ ). Сравнение можно провести при помощи графика зависимости  $\langle n \rangle / \langle n_0 \rangle$  от  $\eta$  (рис. 9). На данном графике отчётливо наблюдается монотонное убывание отношения  $\langle n \rangle / \langle n_0 \rangle$ , что свидетельствует об ослаблении влияния увеличения количества струн на прирост множественности при учёте слияния струн.

### 3.2.3 Вычисление корреляционного коэффициента

Вычисления на данном этапе (четвёртом) проводятся с использованием параболического распределения (формула 2), а область взаимодействия фиксируется как единичная окружность. Значения множественности  $n_F$  и  $n_B$  для каждого события подставляются в формулу 6, затем строится график зависимости  $b(\eta)$ . Полученные графики представлены на рис. 10 и 11 (отмасштабирован по оси  $b$  для наглядности) в приложении; значения  $\eta \approx 2.80$  соответствует энергиям RHIC, а  $\eta \approx 10.35$  – LHC. Нетрудно заметить, что учёт кластеризации слабо влияет на поведение  $b$ : количественно различие  $b_{\text{fusion}}$  и  $b_{\text{no fusion}}$  можно выразить средним процентным соотношением  $\langle D \rangle$

$$\langle D \rangle = \left\langle 1 - \frac{\min(b_{\text{fusion}}, b_{\text{no fusion}})}{\max(b_{\text{fusion}}, b_{\text{no fusion}})} \right\rangle, \quad (7)$$

усреднение ведётся по всем  $\eta$ ; тогда для проведённых симуляций  $\langle D \rangle = 1.57\%$ . Интересно отметить, что при слиянии струн изгиб графика  $b(\eta)$  становится менее пологим, нежели у графика  $b(\eta)$  без слияния.

В качестве проверки правильности вычисления  $b$  программой можно рассмотреть график зависимости  $b(\eta)$  при симуляциях с  $N = \langle N \rangle$ , т.е. вместо равномерно распределённого значения  $N$  взять среднее. Ожидается, что в таком случае коррелятор  $b$  будет близок к нулю, так как при расчётах с фиксированным  $N$  корреляции наблюдать невозможно, и числитель в формуле 6 становится малым. Действительно, первое слагаемое в числителе можно расписать как

$$\langle n_F n_B \rangle = \langle n_F \rangle \langle n_B \rangle + K_{n_F n_B}, \quad K_{n_F n_B} = \langle (n_F - \langle n_F \rangle)(n_B - \langle n_B \rangle) \rangle,$$

где  $K_{n_F n_B}$  – корреляционный момент случайных величин  $n_F$  и  $n_B$ , тогда второе слагаемое в числителе сократится с первым из выражения  $\langle n_F n_B \rangle$ , и останется лишь  $K_{n_F n_B}$ . При постоянном  $N$  в каждой симуляции величины  $n_F$  и  $n_B$  становятся независимыми, так как генерируются последовательно для одной и той же конфигурации кластеров технически независимо, следовательно

$$K_{n_F n_B} = \langle (n_F - \langle n_F \rangle)(n_B - \langle n_B \rangle) \rangle = \langle n_F - \langle n_F \rangle \rangle \langle n_B - \langle n_B \rangle \rangle = 0.$$

Из графика зависимости  $b(\eta)$  (симуляции проведены с учётом слияния струн) в приложении (рис. 12 и 13) видно, что на практике вышеуказанное предположение выполняется.

Вычисления  $b(\eta)$  в случае со слиянием струн занимали приблизительно 2 часа для 10000 симуляций, а в случае без слияния – около 2 минут, тоже для 10000. Столь большая разница в скоростях расчёта связана с отсутствием необходимости во втором случае вводить граф пересечений струн и пресчитывать кластеры.

## 4 Заключение

Феноменологическая модель кварк-глюонных струн неоднократно доказывала свою превосходную адекватность описания хромодинамических процессов, происходящих в мягкой области рассеяния ультрарелятивистских ядер. Её важность для более глубокого понимания принципов Стандартной модели неоспорим, так как она хорошо описывает взаимодействие одного кварк-глюонного поля с другими.

В данной работе приведены результаты модельного теоретического анализа рассеяний ультрарелятивистских ядер при высоких энергиях порядка RHIC (200 ГэВ), LHC (13 ТэВ). Для получения этих результатов была постепенно, в 4 этапа, разработана специальная программа – генератор событий – моделирующая поперечное сечение столкновения ядер в виде непрерывно распределённых сечений струн фиксированного радиуса, а также проводящая

расчёты множественности рождённых адронов и их корреляций, выражаемых корреляционным коэффициентом  $b$ , в переднем и заднем быстротных окнах с учётом слияния (пересечения) данных струн и без. Результатами являются построенные графики зависимости  $\langle n \rangle / \langle n_0 \rangle$  от  $\eta$  – для проверки ослабления зависимости множественности в одном событии от изменения числа струн в сечении при большом количестве этих струн и зависимости  $b$  от  $\eta$  – для проверки влияния учёта кластеризации струн на ФВ-корреляции множественности; величина  $\eta$  играет роль средней плотности струн в области взаимодействия. Из вида этих графиков можно сделать два вывода:

1. При учёте слияния струн усреднённая по событиям при фиксированном  $\eta$  множественность адронов действительно получает более слабый прирост при увеличении плотности струн на больших значениях  $\eta$  (таких, что почти вся область взаимодействия покрыта струнами, слившимися в один кластер).
2. Корреляционный коэффициент  $b$ , как и ожидалось, слабо зависит от введения учёта слияния струн: количественное различие как среднее процентное соотношение составляет  $\langle D \rangle = 1.57\%$ .

Таким образом, основные поставленные задачи были выполнены.

Разработка генератора на текущий момент ещё не закончена. В перспективе – более оптимизированный код; подготовка программы к расчётам  $p_t p_t$  и  $p_t n$  корреляций, представляющих больший интерес, чем  $nn$ . Также планируется использование более реалистичных распределений струн (например, потенциал Вудса-Саксона), введение переменной центральности столкновений, а также уменьшение дисперсии флуктуаций  $N$  в симуляции.

## 5 Список литературы

- [1] D. Perret-Gallix (2013). *Computational Particle Physics for Event Generators and Data Analysis*. J. Phys. Conf. Ser. **454**. 012051 arXiv:1301.1211
- [2] L. Taylor (2012). *Observation of a New Particle with a Mass of 125 GeV*. CMS Experiment, CERN <http://cms.web.cern.ch/news/observation-new-particle-mass-125-gev>
- [3] F.J. Yndurain (1993). *Quantum Chromodynamics: An Introduction to the Theory of Quarks and Gluons*. ISBN 978-3-540-33210-7
- [4] S. Sapeta (2016). *QCD and Jets at Hadron Colliders*. Prog. Part. Nucl. Phys. **89**. 1-55 arXiv:1511.09336
- [5] E. Nurse (2011). *Soft-QCD at Hadron Colliders* (slides). [http://www.hep.ucl.ac.uk/~mw/Post\\_Grads/2011-12/SoftQCDLecture.pdf](http://www.hep.ucl.ac.uk/~mw/Post_Grads/2011-12/SoftQCDLecture.pdf)
- [6] D.S. Cerci (2017). *Soft and Hard QCD Processes in CMS* (slides). [https://indico.cern.ch/event/614845/contributions/2728799/attachments/1529660/2398415/13\\_DSunarCerci.pdf](https://indico.cern.ch/event/614845/contributions/2728799/attachments/1529660/2398415/13_DSunarCerci.pdf)
- [7] A.A. Kaidalov (1982). *The quark-gluon structure of the pomeron and the rise of inclusive spectra at high energies*. Phys. Lett. B **116**. 459-463 [https://doi.org/10.1016/0370-2693\(82\)90168-X](https://doi.org/10.1016/0370-2693(82)90168-X)
- [8] A.A. Kaidalov, K.A. Ter-Martirosyan (1982). *Pomeron as quark-gluon strings and multiple hadron production at SPS-Collider energies*. Phys. Lett. B **117B**. 247-251 [https://doi.org/10.1016/0370-2693\(82\)90556-1](https://doi.org/10.1016/0370-2693(82)90556-1)

- [9] A. Capella, U. Sukhatme, Tan Chung-I, J. Tran Thanh Van (1979). *Jets in small- $p_T$  hadronic collisions, universality of quark fragmentation, and rising rapidity plateaus*. Phys. Lett. B **81**. 68-74 [https://doi.org/10.1016/0370-2693\(79\)90718-4](https://doi.org/10.1016/0370-2693(79)90718-4)
- [10] A. Capella, U. Sukhatme, Tan Chung-I, J. Tran Thanh Van (1994). *Dual parton model*. Phys. Rep. **236**. 225-329 [https://doi.org/10.1016/0370-1573\(94\)90064-7](https://doi.org/10.1016/0370-1573(94)90064-7)
- [11] G.I. Lykasov, M.N. Sergeenko (1996). *Semihard Hadron Processes and Quark-gluon String Model*. Z.Phys. **C70**. 455-462 arXiv:hep-ph/9502316
- [12] T.S. Biro, H.B. Nielsen, J. Knoll (1984). *Colour rope model for extreme relativistic heavy ion collisions*. Nucl. Phys. B **245**. 449-468 [https://doi.org/10.1016/0550-3213\(84\)90441-3](https://doi.org/10.1016/0550-3213(84)90441-3)
- [13] M.A. Braun and C. Pajares (1992). *Particle production in nuclear collisions and string interactions*. Phys. Lett. B **f287**. 154
- [14] M.A. Braun and C. Pajares (1993). *A probabilistic model of interacting strings*. Nucl. Phys. B **f390**. 542
- [15] N.S. Amelin, N. Armesto, M.A. Braun, E.G. Ferreira and C.Pajares (1994). *Long and Short Range Correlations: A Signature of String Fusion*. Phys. Rev. Lett. **f73** 2813
- [16] V.V. Vechernin, C. Pajares, M.A. Braun (2001). *Forward-backward multiplicity correlations, low  $p_T$  distributions in the central region and the fusion of colour strings*. CERN, Geneva. ALICE Collaboration. <http://cds.cern.ch/record/689450>
- [17] В.В. Вечернин, Р.С. Колеватов (2007). *О корреляциях множественности и  $p_t$  в столкновениях ультрарелятивистских ионов*. Ядерная физика **70** 1846-1857 <http://elibrary.ru/item.asp?id=9549732>
- [18] В.В. Вечернин, Р.С. Колеватов (2007). *Дальние корреляции между поперечными импульсами заряженных частиц в релятивистских ядерных столкновениях* Ядерная физика **70** 1858-1867 <http://elibrary.ru/item.asp?id=9549733>
- [19] S. Weinzierl (2000). *Introduction to Monte Carlo methods*. Topical lectures given at the Research School Subatomic Physics, Amsterdam, June 2000 arXiv:hep-ph/0006269
- [20] N. Srimanobhas (2010). *Introduction to Monte Carlo for Particle Physics Study* (slides). <https://indico.cern.ch/event/92209/contributions/2114409/attachments/1098701/1567290/CST2010-MC.pdf>
- [21] V.V. Vechernin, R.S. Kolevatov (2004). *Cellular Approach to Long-Range  $p_t$  and Multiplicity Correlations in the String Fusion Model*. Vestnik SPbU **ser.4, no.4**. 11-27 arXiv:hep-ph/0305136
- [22] V.V. Vechernin, R.S. Kolevatov (2004). *Simple Cellular Model of Long-Range Multiplicity and  $p_t$  Correlations in High-Energy Nuclear Collisions*. Vestnik SPbU **ser.4, no.2**. 12-23 arXiv:hep-ph/0305136
- [23] M.A. Braun, C. Pajares (2000) *Implication of percolation of colour strings on multiplicities, correlations and the transverse momentum*. Eur. Phys. J. **C16**. 349–359 arXiv:hep-ph/9907332
- [24] J. Dias de Deus, A.S. Hirsch, C. Pajares, et al. (2012). *Clustering of color sources and the shear viscosity of the QGP in heavy ion collisions at RHIC and LHC energies*. Eur. Phys. J. C **72**:2123 <https://doi.org/10.1140/epjc/s10052-012-2123-x>
- [25] D. Kharzeev, M. Nardi (2001). *Hadron production in nuclear collisions at RHIC and high density QCD*. Phys. Lett. B **507** 121-128 arXiv:nucl-th/0012025

## 6 Приложение

### 6.1 Изображения

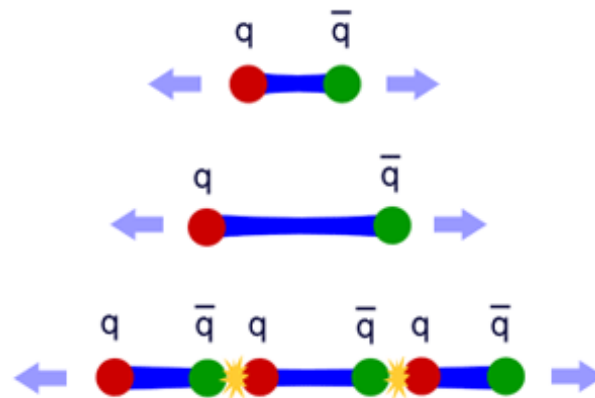


Рис. 1: Схема фрагментации кварк-глюонной струны при её натяжении с последующей адронизацией (прим.: разные цвета использованы не с целью изобразить цветовой заряд, а лишь для наглядности)

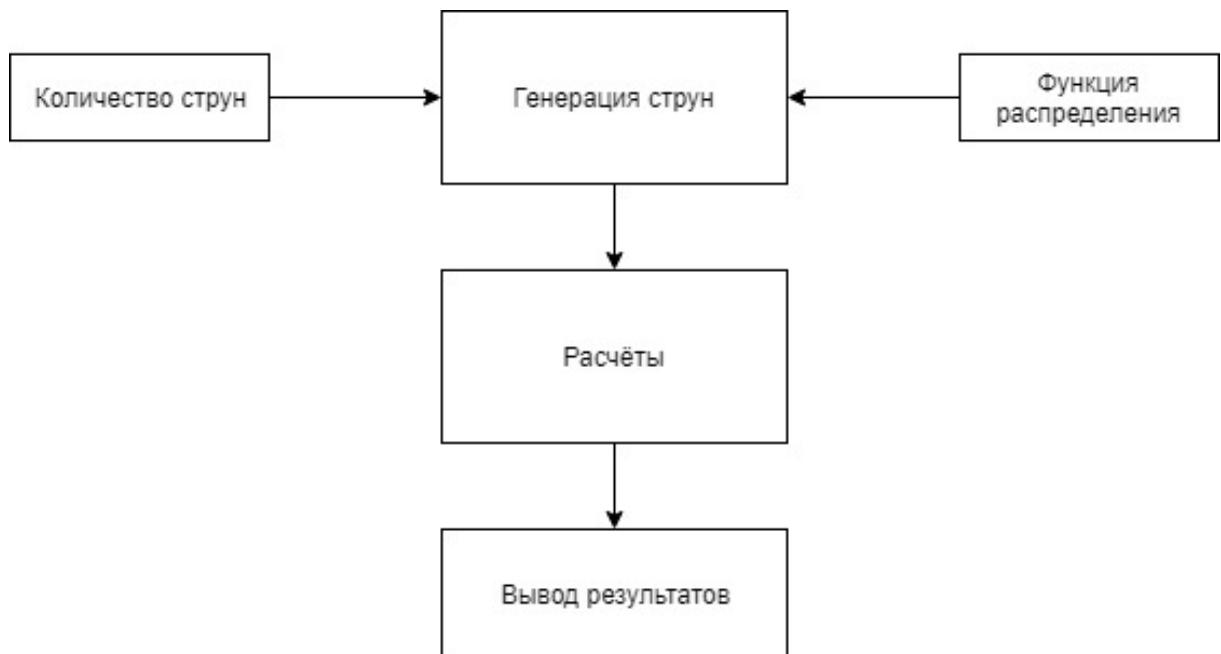


Рис. 2: Принципиальная схема работы генератора

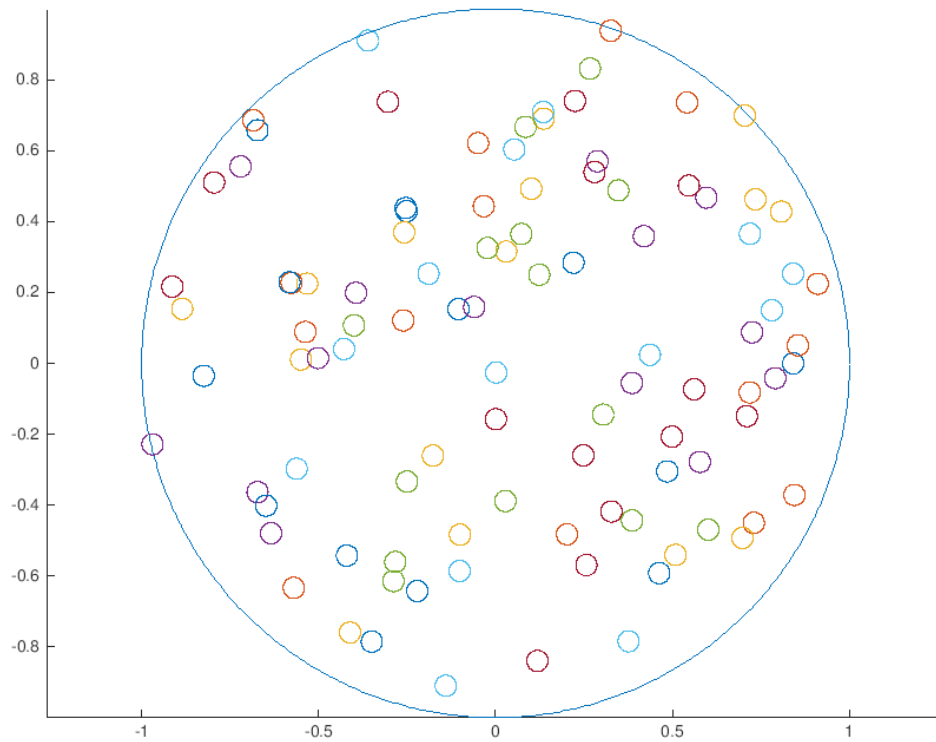


Рис. 3: Визуализация разыгрывания 100 струн при равномерном распределении в области единичного круга. Отношение радиуса круговой области к радиусу генерируемых струн примерно соответствует отношению радиуса реального ядра к радиусу реальных струн в поперечном сечении (7.5 фм к  $0.2 \div 0.3$  фм соответственно)

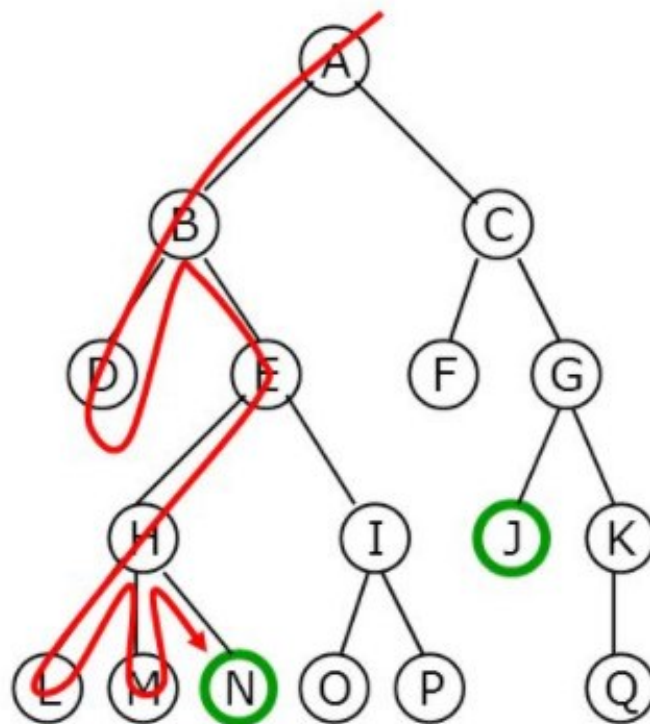


Рис. 4: Схема обхода графа методом Depth-first search

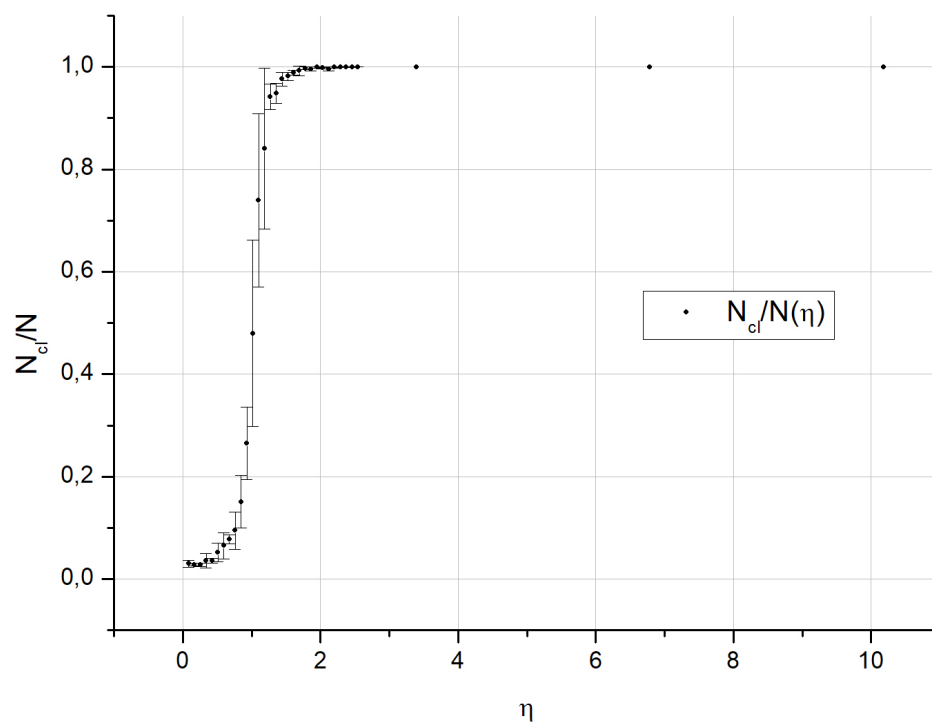


Рис. 5: График зависимости  $N_{cl}/N$  от  $\eta$  (MATLAB, усреднено по 5 симуляциям)

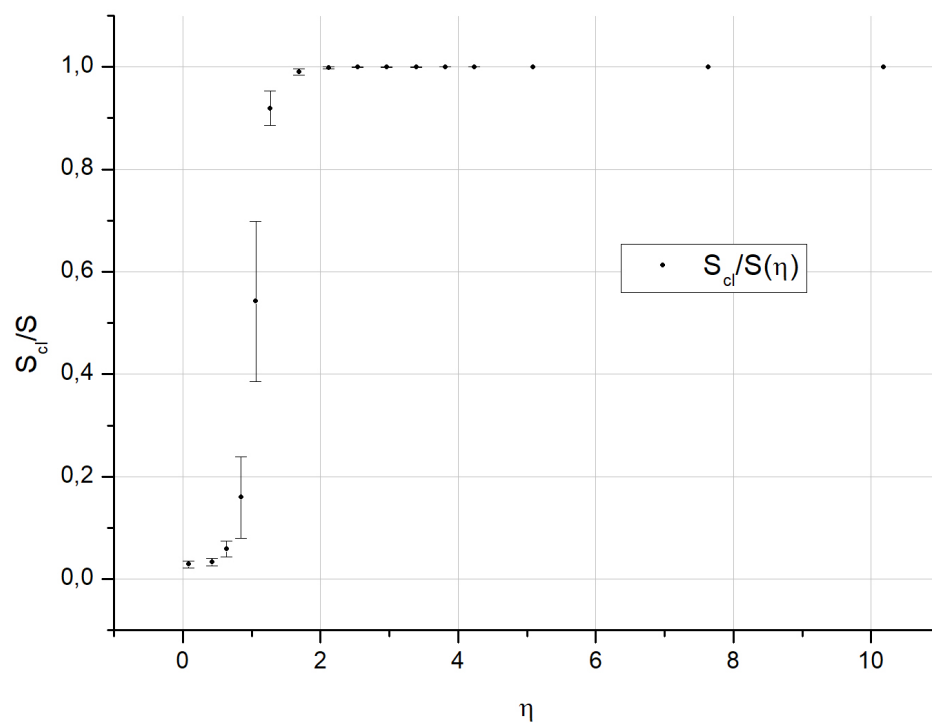


Рис. 6: График зависимости  $S_{cl}/S$  от  $\eta$  (MATLAB, усреднено по 60 симуляциям)

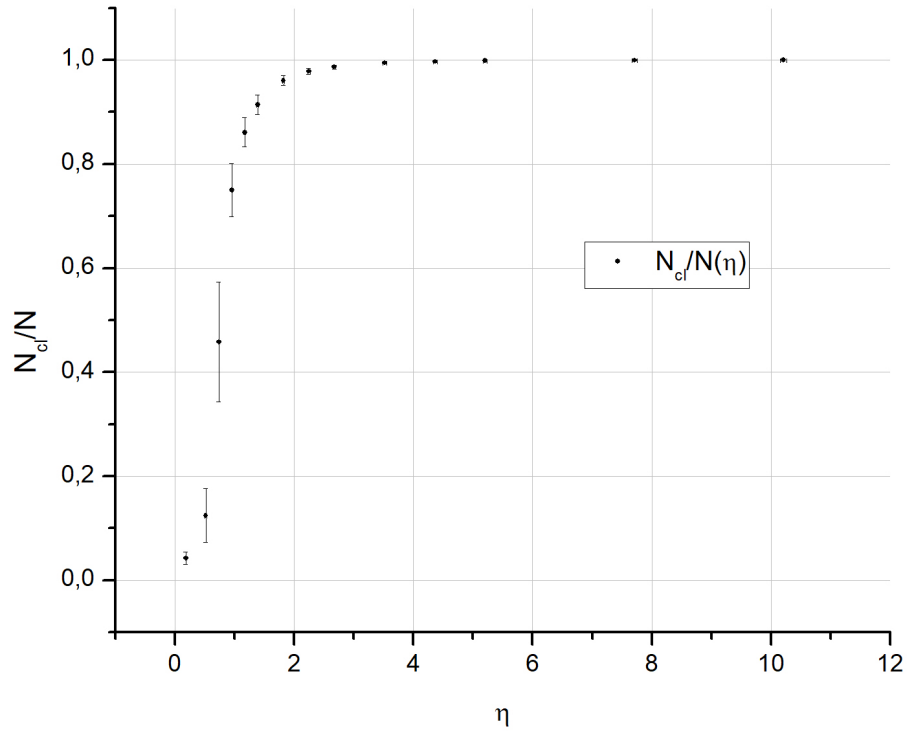


Рис. 7: График зависимости  $N_{cl}/N$  от  $\eta$  (C++, усреднено по 3638 симуляциям)

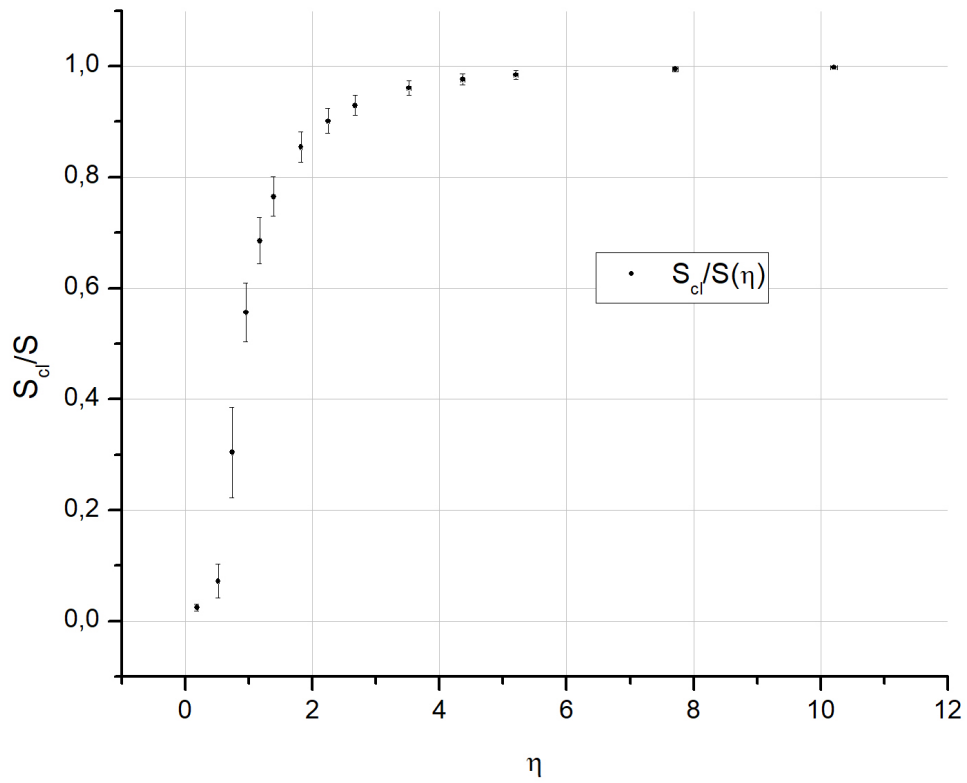


Рис. 8: График зависимости  $S_{cl}/S$  от  $\eta$  (C++, усреднено по 3638 симуляциям)



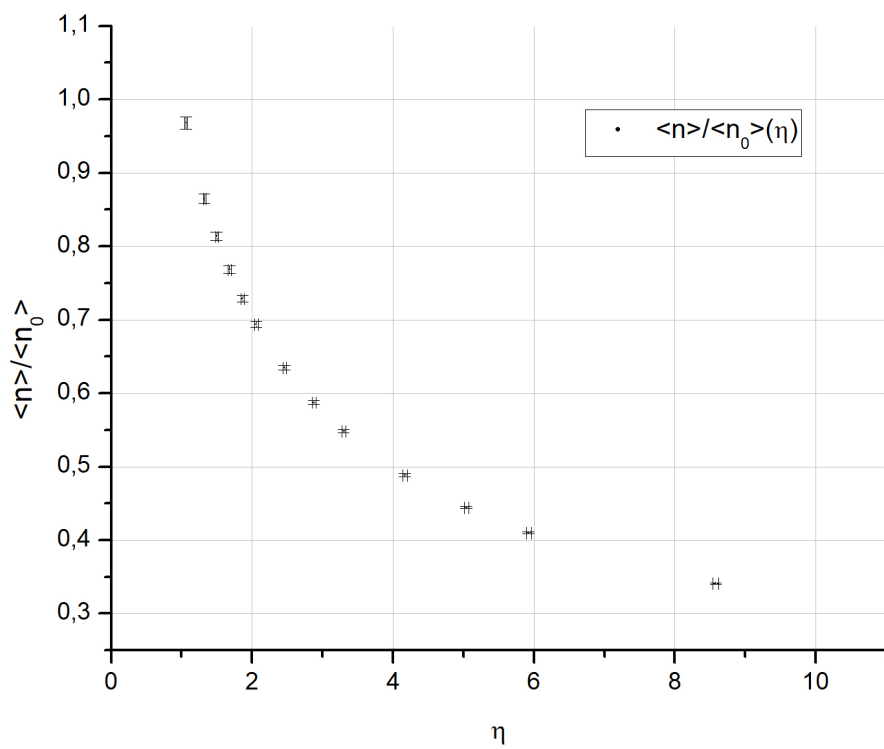


Рис. 9: График зависимости  $\langle n \rangle / \langle n_0 \rangle$  от  $\eta$  (C++, усреднено по 3563 симуляциям)

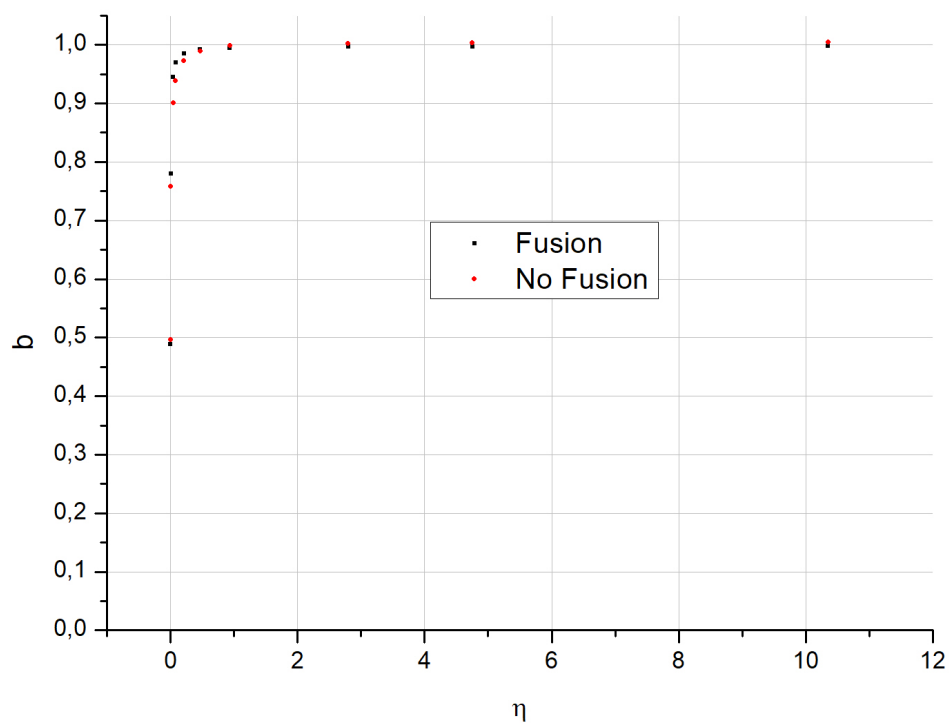


Рис. 10: График зависимости  $b$  от  $\eta$  с учётом (Fusion) и без учёта (No Fusion) слияния струн (C++, 10000 симуляций)

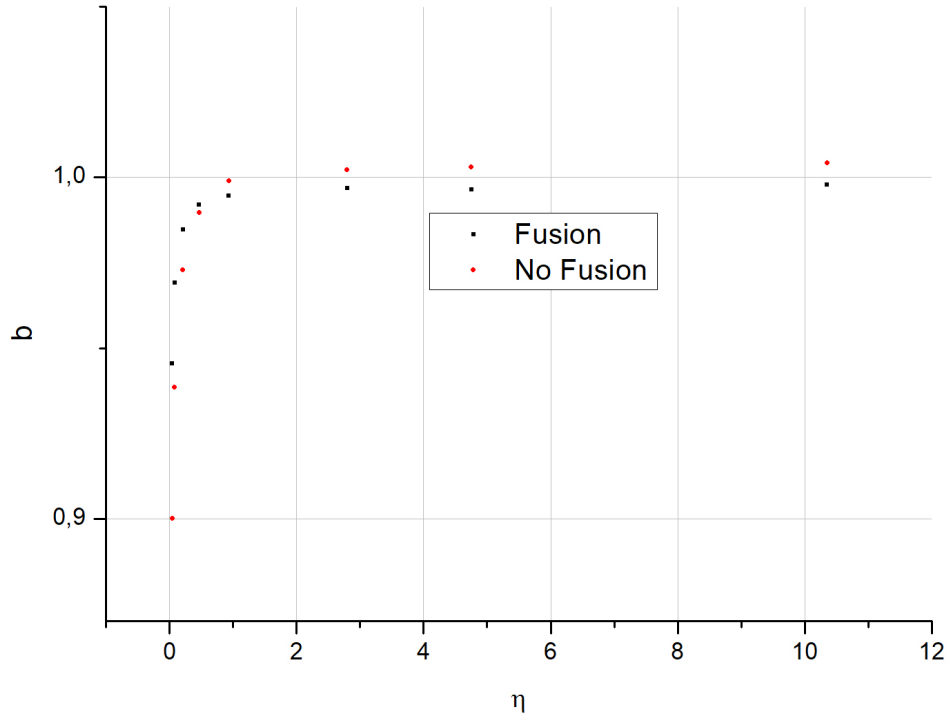


Рис. 11: Отмасштабированный по оси  $b$  график зависимости  $b$  от  $\eta$  с учётом (Fusion) и без учёта (No Fusion) слияния струн (C++, 10000 симуляций)

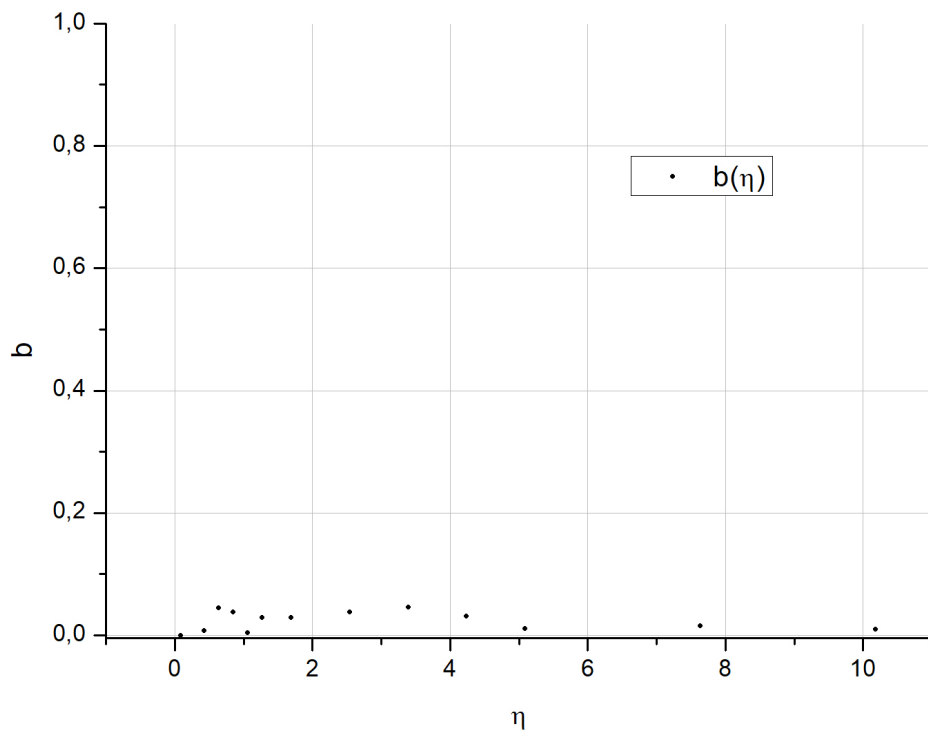


Рис. 12: График зависимости  $b$  от  $\eta$  при  $N = \langle N \rangle$  со слиянием (C++, 10000 симуляций)

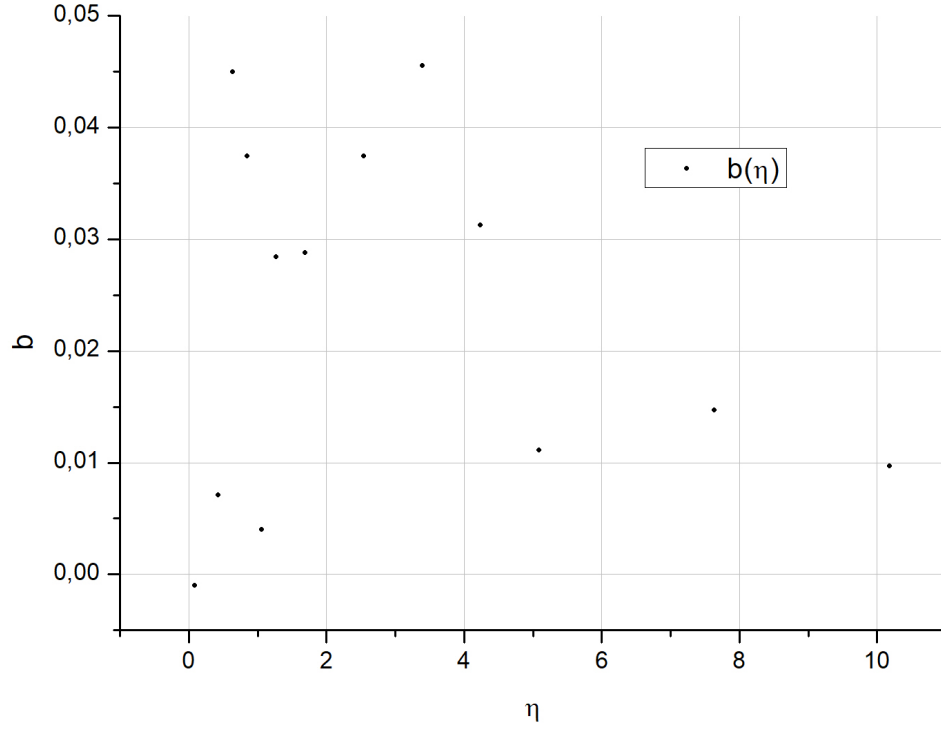


Рис. 13: Отмасштабированный по оси  $b$  график зависимости  $b$  от  $\eta$  при  $N = \langle N \rangle$  со слиянием (C++, 10000 симуляций)

## 6.2 Таблицы

$\eta$	$N_{cl}/N$	$\Delta(N_{cl}/N)$	$\eta$	$N_{cl}/N$	$\Delta(N_{cl}/N)$	$\eta$	$N_{cl}/N$	$\Delta(N_{cl}/N)$
0.085	0.030	0.007	1.018	0.480	0.182	1.951	0.999	0.001
0.170	0.028	0.003	1.103	0.739	0.170	2.036	0.998	0.001
0.255	0.028	0.004	1.188	0.840	0.157	2.121	0.996	0.004
0.339	0.036	0.014	1.273	0.941	0.025	2.206	0.999	0.001
0.424	0.036	0.005	1.357	0.948	0.020	2.291	1.000	0.001
0.509	0.052	0.018	1.442	0.976	0.013	2.375	1.000	0
0.594	0.065	0.026	1.527	0.982	0.008	2.460	0.999	0
0.679	0.077	0.009	1.612	0.989	0.004	2.545	1.000	0
0.764	0.095	0.037	1.697	0.992	0.009	3.393	1.000	0
0.848	0.151	0.051	1.782	0.997	0.003	6.787	1.000	0
0.933	0.265	0.071	1.866	0.996	0.004	10.18	1.000	0

Таблица 1: Значения  $\eta$  и соответствующие  $N_{cl}/N$  с погрешностями  $\Delta(N_{cl}/N)$  к рис. 5

$\eta$	$S_{cl}/S$	$\Delta(S_{cl}/S)$	$\eta$	$S_{cl}/S$	$\Delta(S_{cl}/S)$
0.085	0.029	0.007	2.545	0.999	0.001
0.424	0.033	0.008	2.969	1.000	0
0.636	0.058	0.016	3.393	1.000	0
0.848	0.159	0.079	3.818	1.000	0
1.060	0.542	0.156	4.242	1.000	0
1.273	0.919	0.034	5.090	1.000	0
1.697	0.990	0.006	7.635	1.000	0
2.121	0.998	0.002	10.18	1.000	0

Таблица 2: Значения  $\eta$  и соответствующие  $S_{cl}/S$  с погрешностями  $\Delta(S_{cl}/S)$  к рис. 6

$\eta$	$N_{cl}/N$	$\Delta\eta$	$\Delta(N_{cl}/N)$	$\eta$	$N_{cl}/N$	$\Delta\eta$	$\Delta(N_{cl}/N)$
0.186	0.043	0.008	0.012	2.249	0.978	0.023	0.006
0.520	0.124	0.011	0.052	2.676	0.987	0.025	0.004
0.739	0.458	0.013	0.115	3.523	0.994	0.029	0.002
0.957	0.750	0.014	0.051	4.366	0.997	0.033	0.001
1.175	0.861	0.016	0.028	5.206	0.998	0.036	0.001
1.391	0.914	0.018	0.019	7.714	1.000	0.045	0
1.822	0.960	0.020	0.009	10.212	1.000	0.054	0

Таблица 3: Значения  $\eta$  и соответствующие  $N_{cl}/N$  с погрешностями  $\Delta\eta$  и  $\Delta(N_{cl}/N)$  к рис. 7

$\eta$	$S_{cl}/S$	$\Delta\eta$	$\Delta(S_{cl}/S)$	$\eta$	$S_{cl}/S$	$\Delta\eta$	$\Delta(S_{cl}/S)$
0.186	0.025	0.008	0.007	2.249	0.901	0.023	0.022
0.520	0.072	0.011	0.031	2.676	0.93	0.025	0.018
0.739	0.304	0.013	0.081	3.523	0.961	0.029	0.013
0.957	0.557	0.014	0.053	4.366	0.976	0.033	0.010
1.175	0.686	0.016	0.042	5.206	0.984	0.036	0.008
1.391	0.765	0.018	0.035	7.714	0.995	0.045	0.004
1.822	0.854	0.020	0.027	10.212	0.998	0.054	0.003

Таблица 4: Значения  $\eta$  и соответствующие  $S_{cl}/S$  с погрешностями  $\Delta\eta$  и  $\Delta(S_{cl}/S)$  к рис. 8

$\eta$	$\langle n \rangle / \langle n_0 \rangle$	$\Delta\eta$	$\Delta(\langle n \rangle / \langle n_0 \rangle)$	$\eta$	$\langle n \rangle / \langle n_0 \rangle$	$\Delta\eta$	$\Delta(\langle n \rangle / \langle n_0 \rangle)$
1.063	0.968	0.017	0.008	2.879	0.587	0.027	0.003
1.326	0.865	0.019	0.007	3.304	0.549	0.028	0.002
1.499	0.814	0.020	0.006	4.169	0.489	0.031	0.002
1.680	0.769	0.022	0.005	5.044	0.445	0.033	0.002
1.868	0.729	0.023	0.005	5.926	0.410	0.036	0.001
2.061	0.694	0.024	0.004	8.581	0.341	0.043	0.001
2.463	0.635	0.025	0.003	11.236	0.298	0.052	0.001

Таблица 5: Значения  $\eta$  и  $\langle n \rangle / \langle n_0 \rangle$  с погрешностями  $\Delta\eta$  и  $\Delta(\langle n \rangle / \langle n_0 \rangle)$  к рис. 9

$\eta$	$b_{\text{fusion}}$	$b_{\text{no fusion}}$	$\eta$	$b_{\text{fusion}}$	$b_{\text{no fusion}}$
0.002	0.489	0.496	0.467	0.992	0.990
0.008	0.780	0.758	0.933	0.995	0.999
0.042	0.945	0.900	2.800	0.997	1.002
0.085	0.969	0.938	4.751	0.996	1.003
0.212	0.985	0.973	10.35	0.998	1.004

Таблица 6: Значения  $\eta$  и соответствующие  $b_{\text{fusion}}$  и  $b_{\text{no fusion}}$  к рис. 10 и рис. 11

$\eta$	$b_{\text{fusion}}$	$\eta$	$b_{\text{fusion}}$
0.085	-0.001	2.545	0.037
0.424	0.007	3.393	0.046
0.636	0.045	4.242	0.031
0.848	0.037	5.090	0.011
1.060	0.004	7.635	0.015
1.273	0.028	10.180	0.010
1.697	0.029	-	-

Таблица 7: Значения  $\eta$  и соответствующие  $b_{\text{fusion}}$  к рис. 12 и рис. 13

## 6.3 Код программы

Листинг 1: Основные методы класса “Simulation”

```
void f_nGen(); // 1
void f_GenerateXY_Neym(); // 2
void f_FillGraph(); // 3
void f_FindConnComp(); // 4
void f_FindMaxComp(); // 5
void f_FindNN(); // 6
void f_FindSS(); // 7
void f_FindMulFB(); // 8
void f_FindMulRatio(); // 9
void f_WriteData(); // 10
```

По порядку: 1 – генерация значения  $N$  через заданное среднее  $\langle N \rangle$ ; 2 – генерация координат струн методом Неймана; 3 – создание графа пересекающихся струн; 4 – поиск компонент связности графа; 5 – поиск самого большого кластера по количеству струн; 6 – расчёт отношения количества струн в максимальном кластере к общему количеству разбросанных струн; 7 – расчёт отношения площади максимального кластера к общей площади взаимодействия; 8 – генерация множественности вперёд-назад с учётом слияния струн и без; 9 – генерация множественности с учётом слияния струн и без; 10 – запись данных в файл для последующей обработки.

Листинг 2: Вспомогательные методы класса “Simulation”

```
void f_bCalc(); // 1
void f_dfs(unsigned short int); // 2
unsigned short int f_in_PDF_N(unsigned short int); // 3
float f_in_PDF(float, float); // 4
float f_in_distXY(float, float, float, float, unsigned short int); // 5
```

По порядку: 1 – расчёт корреляционного параметра  $b$  и запись его в файл; 2 – функция обхода графа для поиска компонент связности методом DFS (Depth-first search); 3 – метод, хранящий функцию распределения (или генератор) значения  $N$  через заданное среднее  $\langle N \rangle$ ; 4 – метод, хранящий функцию распределения разброса струн в области взаимодействия; 5 – метод, возвращающий расстояние между двумя заданными точками.

Листинг 3: Поиск компонент связности с помощью DFS

```
void Simulation::f_FindConnComp()
{
    v_compData.clear(); // clear the vcm
    v_comp.clear(); // clear the temporary cluster array
    v_used.clear(); // clear the trigger marks
    for(usint i = 0; i < N; ++i)
        v_used.push_back(false); // make all the coordinates unused
    for(usint i = 0; i < N; ++i)
        if(!v_used[i]) // if unused...
        {
            v_comp.clear();
            f_dfs(i); // go depth-first search
            if(!v_comp.empty())
            {
                v_compData.resize(i + 1);
                v_compData[i] = v_comp; // write into the vcm
            }
        }
}

void Simulation::f_dfs(usint v)
{
    v_used[v] = true; // string v searching for nearby strings
```

```

v_comp.push_back(v); // push v'th string to a component being made
for(uint i = 0; i < N; ++i)
{
    if(g_connGraph[v][i] < 65535) // 65535 is default (if uint is used)
    {
        uint to = g_connGraph[v][i]; // get a neighbor from the graph
        if(! v_used[to]) // check if a neighbor was not checked yet
            f_dfs(to); // recursively repeat dfs
    }
}
}

```