

ФЕДЕРАЛЬНОЕ АГЕНТСТВО ПО ОБРАЗОВАНИЮ

Государственное образовательное учреждение
высшего профессионального образования
"Московский Физико-Технический институт
(Государственный университет)"

Кафедра теоретической астрофизики и квантовой теории поля

Решеточное изучение свойств глюодинамики во внешнем хромоманнитном поле

(Выпускная квалификационная работа на степень бакалавра)

Выполнила:

студентка 624 группы
Петрова Елена Владимировна

Научный руководитель:

к.ф.-м.н., Котов А.Ю.

Долгопрудный

2020

Содержание

1	Введение	2
2	Моделирование на решетке	5
3	Абелево хромоманнитное поле на решетке	8
3.1	Предельный случай	8
3.2	Хромоманнитное поле. Способ 1 (матричный)	8
3.3	Хромоманнитное поле. Способ 2 (угловой)	9
4	Характеристики вычислений	11
5	Измеренные величины	11
6	Полученные результаты	12
6.1	U(1) теория	12
6.2	SU(2) теория	12
7	Заключение	16
	Приложение	18

1 Введение

Понимание механизма конфайнмента до сих пор остается одной из главных проблем физики высоких энергий, а изучение кварк-глюонной плазмы, возникающей в результате деконфайнмента, одним из основных направлений исследований.

Глюодинамика, теория, описывающая глюоны, очень похожа на электродинамику. Различие заключается в калибровочной группе: для глюодинамики калибровочная группа неабелева. Вследствие этого, аналоги электрического и магнитного полей и тензор напряженности в SU(2) теории выглядят немного по-другому. Поля имеют добавку в виде коммутатора и принимают следующий вид:

$$\begin{aligned} F_{\mu\nu}^a &= \partial_\mu A_\nu^a - \partial_\nu A_\mu^a - f^{abc} A_\mu^b A_\nu^c \\ F_{\mu\nu} &= F_{\mu\nu}^a T_a \\ \vec{E} &= -\nabla\phi - \frac{\partial\vec{A}}{\partial t} + i[\phi, \vec{A}] \\ \vec{H} &= \text{rot}\vec{A} + i[\vec{A}, \vec{A}] \end{aligned} \quad (1)$$

Где по a подразумевается суммирование во второй формуле, T_a - генераторы соответствующей алгебры, а $\varphi = A_0^a T_a$. Такие поля называются хромоелектрическими и хромагнитными соответственно. А действие без фермионов имеет вид

$$S = \frac{1}{4g^2} \sum_{a=1}^8 \int F_{\mu\nu}^a{}^2 \quad (2)$$

Существенным отличием от электродинамики является то, что константа связи g порядка единицы при малых импульсах, следовательно, нельзя применить пертурбативный метод. Это значительно усложняет исследование КХД при низких энергиях.

Интересным свойством КХД при низких температурах является линейно растущий потенциал. Следствием этого является невозможность наблюдения отдельных кварков, так как невозможно разорвать связь между ними. В этом случае, наблюдаемыми являются адроны - бесцветные частицы, состоящие из нескольких кварков. Это явление и называется конфайнментом. При температурах выше критической, система переходит в другое состояние - фазу деконфайнмента или так называемую кварк-глюонную плазму. Потенциал между частицами становится достаточно маленьким, что становится возможным наблюдение кварков по отдельности.

Приблизительное изображение фазовой диаграммы конфайнмент-деконфайнмент изображено на Рис. 1 Хорошо исследована на компьютерах та часть диаграммы, которая соответствует нулевой барионной плотности [1].

Другим важным местом на графике является область ненулевой барионной плотности. Предел большой барионной плотности имеет место в физике нейтронных звезд. Считается, что в центре таких звезд может находиться именно кварк-глюонная плазма. Кроме того именно этот предел реализуется при столкновении тяжелых ионов и наиболее приближен к эксперименталь-

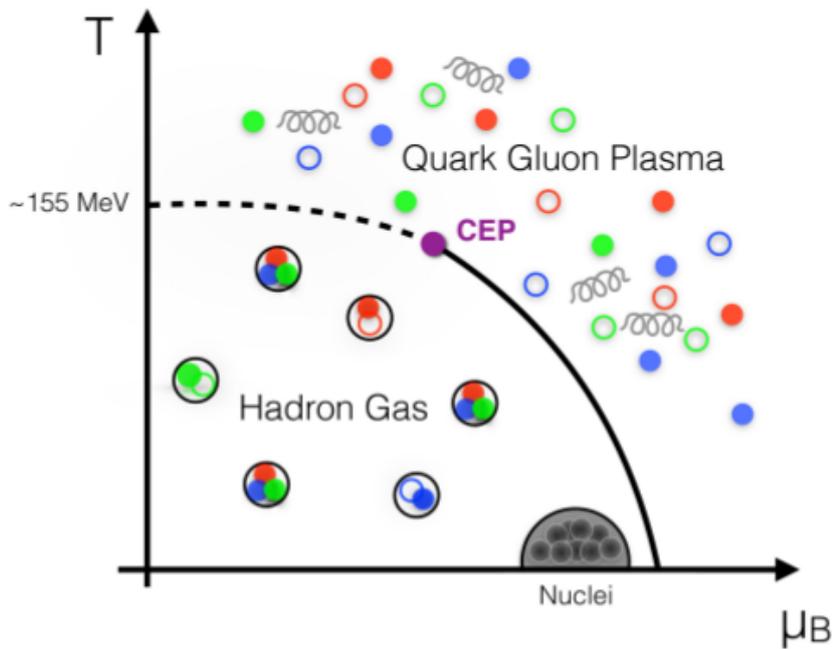


Рис. 1: Фазовая диаграмма КХД

ной модели.

На данный момент не существует математически точного решения проблемы конфайнмента, однако значительные продвижения были сделаны благодаря компьютерам. В том числе, решеточная КХД подтвердила наличие конфайнмента.

В настоящее время, как теоретики, так и экспериментаторы, заинтересованы в поисках новых эффектов в кварк-глюонной плазме, в которой кварки и глюоны, носители цветового заряда, представляют собой самостоятельные степени свободы. Внимание ученых было уделено и влиянию внешнего хромагнитного поля на свойства хромодинамики. В случае абелева поля $A_\mu^1 = A_\mu^2 = 0$, и, следовательно, член со структурной константой обращается в 0. Такое поле легче исследовать на решетке, нежели неабелево.

Влияние хромагнитного поля на КХД было рассмотрено в разных аспектах. Так, в статье [2] было описано спонтанное возникновение хромагнитного поля при высоких температурах в $SU(2)$ глюодинамике. Авторы заметили, что в фазе деконфайнмента возникают хромагнитные поля, сила которых зависит от температуры системы. Измеряя свободную энергию в зависимости от поля, были получены данные, указывающие на минимум энергии при ненулевом поле, из чего были сделаны выводы, что такие поля должны возникать в системе.

Интересные данные были получены в статье [3]. Авторы измерили влияние хромагнитного поля на температуру деконфайнмента в КХД с двумя легкими кварками. Полученные в статье результаты показали, что температура фазового перехода уменьшается с увеличением внешнего хромагнитного поля, более того, увеличение приложенного внешнего поля приводит к увеличению кирального конденсата. Интересно заметить, что в другой статье тех же авторов,

для различных калибровочных теорий без фермионов была измерена зависимость критической температуры от внешнего хромомангнитного поля [4]. В обеих статьях влияние внешнего абелева хромомангнитного поля проявляется одинаково - критическая температура фазового перехода уменьшается с увеличением приложенного внешнего поля.

Более приближенные к экспериментам результаты могут быть получены при изучении столкновений тяжелых ионов. Получающаяся вследствие столкновений глазма, предшественник кварк-глюонной плазмы, содержит в себе сильные хромомангнитные и хромоелектрические поля. В работе [5] было исследовано столкновение частиц в рамках теории конденсата цветового стекла - теоретической модели, описывающей сильное взаимодействие в условиях высоких плотностей. Конфигурации полей, возникающих сразу после столкновения, предполагают возникновение большой плотности заряда Черна-Саймонса и, таким образом, эволюция глазмы может быть описана как распад этого заряда.

Влияние поля пытаются изучать и теоретически, в рамках Nambu-Jona-Lasinio - эффективной модели квантовой хромодинамики [6]. Утверждается, что расчеты в этой модели в некоторых случаях совпадают с решеточным моделированием качественно, а иногда и численно. Авторы взяли зависимость температуры от поля из статьи [4] и применили результаты к изучению влияния динамических кварков на конфайнмент в рамках теории. Полученный результат сравнили с решеточным моделированием и получили, что результаты качественно согласуются со статьей [3]. Однако есть другие места, в которых результаты данной модели не согласуются с моделированием на решетках.

Таким образом, изучение свойств глюодинамики во внешнем хромомангнитном поле может помочь понять, что лежит в основе такого загадочного процесса, как конфайнмент.

В данной работе рассмотрена $SU(2)$ глюодинамика во внешнем абелевом хромомангнитном поле. Целью исследования является изучение влияния хромомангнитного поля на натяжение струны.

2 Моделирование на решетке

Для теории сильных взаимодействий невозможно пертурбативное разложение, поэтому решеточный подход в этой теории является одним из немногих возможных непертурбативных методов исследования.

Рассмотрим ключевые моменты перехода к решеточному формализму. Для большей информации можно обратиться к источникам [7] , [8]. Для решеточного моделирования, первым делом, проводится дискретизация пространства. Координаты в пространстве заменяются на координаты решетки:

$$x_\mu \rightarrow n = (n_1, n_2, n_3, n_4) \quad (3)$$

Шагом решетки a называют расстояние между узлами решетки. А величину $L = aN$ - размер решетки, где N - количество узлов. Рассматривая теорию на решетке, мы делаем одновременно ультрафиолетовое и инфракрасное обрезания по импульсам: длины волн ограничены снизу шагом решетки, а сверху - размером решетки. То есть рассматривая теорию на решетке, мы регуляризуем ее. Для получения результатов непрерывной теории необходимо взять сначала термодинамический ($L \rightarrow \infty$), потом континуальный ($a \rightarrow 0$) пределы.

Заметим, что решеточный подход сохраняет калибровочную инвариантность, в то время как для пертурбативного метода требуется фиксация калибровки. Чтобы определить калибровочно инвариантное действие для $SU(2)$ теории на решетке введем линковые переменные $U_\mu(n)$:

$$U_\mu(n) = \exp[iaA_\mu(n)] \quad (4)$$

Преобразование переменной U введем следующим образом:

$$U_\mu(n) \rightarrow U'_\mu(n) = \Omega(n)U_\mu(n)\Omega(n + \hat{\mu}), \quad \Omega(n) \in SU(N) \quad (5)$$

Такие переменные, которые лежат на ребрах между узлами, можно отождествить с глюонными полями. В непрерывном случае, эти поля соответствуют линии Вильсона, соединяющей эти точки. Более того, поля U являются элементами калибровочной группы.

Простейшей калибровочно инвариантной величиной в такой теории является петля по замкнутому контуру вида:

$$L[U] = \text{Tr} \left[\prod_{n, \mu \in C} U_\mu(N) \right] \quad (6)$$

То есть если применить калибровочное преобразование из формулы 5 и получить, что такая величина остается неизменной под действием калибровочной группы.

$$L[U'] = \text{Tr} \left[\Omega(n_0) \prod_{n, \mu \in C} U_\mu(N) \Omega(n_0)^+ \right] = \text{Tr} \left[\prod_{n, \mu \in C} U_\mu(n) \right] = L[U] \quad (7)$$

Тогда можно рассмотреть самое простое калибровочно инвариантное действие без фермионов:

$$S_W = \beta \sum_n \sum_{\mu < \nu} \left[1 - \frac{1}{N} \text{ReTr} U_{\mu\nu}(n) \right] \quad (8)$$

где $\beta = \frac{2N}{g^2}$, а плакет $U_{\mu\nu}$ имеет вид:

$$U_{\mu\nu}(n) = U_\mu(n) U_\nu(n + \hat{\mu}) U_\mu(n + \hat{\nu})^+ U_\nu(n)^+ \quad (9)$$

Плакет можно изобразить графически

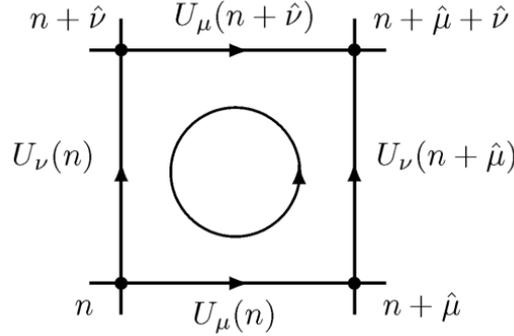


Рис. 2: Плакет

Такое действие называется действием Вильсона. Можно легко проверить, что устремив a к 0 получится хорошо известное непрерывное действие Янга-Милса:

$$S_W \rightarrow \frac{1}{4g^2} \int \text{Tr}[F_{\mu\nu}(x)^2] \quad (10)$$

Наиболее частой задачей в решеточной теории поля является расчет различных наблюдаемых. В теории поля, чтобы найти среднее какой-либо величины, требуется взять континуальный интеграл.

$$\langle O[U] \rangle = \frac{1}{Z} \int D[U] O[U] e^{-S[U]} \quad (11)$$

где $O[U]$ - интересующая наблюдаемая. Решеточный подход решает эту проблему заменив континуальный интеграл на среднее по достаточно большому числу конфигураций, генерируемых с вероятностью, пропорциональной действию.

$$\langle O[U] \rangle = \lim_{N \rightarrow \infty} \frac{1}{N} \sum_{i=1}^N O[U_i] \quad (12)$$

$$dP(U) \sim DU e^{-S[U]}$$

Таким образом, задача сводится к генерации достаточного числа конфигураций с определенной вероятностью. Метод Монте-Карло, представленный различными алгоритмами (алгоритм Метрополиса, "Heatbath" и другие), позволяет это сделать. Полученная марковская цепь конфигураций является основой для измерения наблюдаемой.

Использованный в данной работе алгоритм, "heat bath", включает в себя генерирование

величин x с определенной плотностью вероятности $f(x)$. Имея генератор случайных чисел, это легко осуществить следующим образом. Пусть функция $f(x)$ задана на единичном интервале. Без потери общности можно положить, что она ограничена сверху единицей, тогда совершим следующие действия:

1. С помощью генератора получим пробное число от 0 до 1 для элемента «взвешенного» ряда x .
2. С помощью генератора получим второе число от 0 до 1.
3. Считаем x принятым, если второе число меньше, чем $f(x)$

Повторим эти действия много раз и получим набор величин x , распределенных с плотностью вероятности $f(x)$.

В нашем случае, распределение вероятности нового значения линка имеет вид

$$dp(U) \sim dU \exp\left(\frac{1}{2}\beta \text{Tr}(U \sum_1^6 \bar{U}_\alpha)\right) \quad (13)$$

Где dU - групповая мера, \bar{U}_α произведение трех линков, образующих с U плакету в различных плоскостях. Далее, используем явный вид меры для $SU(2)$:

$$\begin{aligned} SU(2) &= \{a_0 + i\vec{a}\vec{\sigma} \mid a_0^2 + \vec{a}^2 = 1\} \\ dU &\sim da^4 \delta(a^2 - 1) \end{aligned} \quad (14)$$

А так же свойство, заключающееся в том, что сумма элементов группы пропорциональна другому элементу группы:

$$\begin{aligned} \sum_1^6 \bar{U}_\alpha &= k\bar{U} \\ k &= \left| \sum_{\alpha=1}^6 U_\alpha \right|^{\frac{1}{2}} \end{aligned} \quad (15)$$

Далее воспользуемся инвариантностью групповой меры:

$$d(U\bar{U}^{-1}) \sim dU \exp\left(\frac{1}{2}\beta \text{Tr} U\right) = da^4 \delta(a^2 - 1) \exp(\beta k a_0) \quad (16)$$

Таким образом, проблема свелась к генерации точек на трехмерной сфере единичного радиуса с экспоненциальным распределением для a_0 . Тогда, найдя U и \bar{U} , новой переменной на ребре становится $U' = U\bar{U}^{-1}$. Применяв вышеизложенный алгоритм к каждому линку на решетке, получаем новую конфигурацию.

3 Абелево хромагнитное поле на решетке

3.1 Предельный случай

В данной работе рассматривается Вильсоновское действие для $SU(2)$ глюодинамики.

$$S = \beta \sum_n \sum_{\mu < \nu} [1 - \frac{1}{2} Tr U_{\mu\nu}(n)] \quad (17)$$

В непрерывном пределе, мы хотим, чтобы Абелево хромагнитное поле было направлено вдоль оси z , тогда его можно ввести следующим образом:

$$\vec{A}_a^{ext} = \delta_{3,a} \vec{A}^{ext}, \quad A_\mu^{ext} = \delta_{2,\mu} x_1 H \quad (18)$$

Где имеется в виду, что $A_\mu^1 = A_\mu^2 = 0$, а для A_μ^3 не равно 0 только для $\mu = 2$, то есть $A_2^3 = x_1 H$.

Калибровочные поля, в соответствии с записью 18 в предельном случае примут вид:

$$\begin{aligned} U_1^{ext} &= U_4^{ext} = U_3^{ext} = 1 \\ U_2^{ext} &= \sigma_0 \cos \frac{ax_1 H}{2} + i\sigma_3 \sin \frac{ax_1 H}{2} \end{aligned} \quad (19)$$

где σ_0 - единичная матрица, а σ_3 третья матрица Паули.

Теперь необходимо учесть, что на решетке всегда представлено динамическое поле, и нам необходимо учесть взаимодействие внешнего поля с динамическим. Для этого рассмотрим 2 способа, как это можно сделать.

3.2 Хромагнитное поле. Способ 1 (матричный)

Рассмотрим первый способ, описанный в статье [9]. Чтобы ввести постоянное Абелево хромагнитное поле удобно взять плакет в виде:

$$U_{\mu\nu}(n) = \exp[ia^2 F_{\mu\nu}] \quad (20)$$

Рассмотрим влияние внешнего поля на конкретный плакет, тогда изменение будет заметно только в плоскости xu , и преобразование примет вид:

$$U'_{xy} = \exp[ia^2 (H_z + H_z^{ext})] = U_{xy} \exp[ia^2 H] \quad (21)$$

с точностью до коммутатора H_z с H_z^{ext} , где H_z - поле, постоянно представленное на решетке, а H_z^{ext} - внешнее поле, которое мы ввели.

Для вычисления на компьютерах удобнее рассматривать линки, тогда изменение линка во внешнем абелевом хромагнитном поле выглядит следующим образом:

$$U'_y(n + \hat{x}) = U_y(n) \exp(ia^2 H_{ext}) \quad (22)$$

Тогда на решетке можно учесть это поле только на границе:

$$U'_y(0, n_y, n_z, n_t) = U_y(0, n_y, n_z, n_t) e^{i\phi} \quad \phi = a^2 H_{ext} N_x \quad (23)$$

где ϕ есть поток поля через петлю размером $N_x \times 1$, а матрица $e^{i\phi}$ имеет вид:

$$e^{i\phi} = e^{i\varphi_3 \sigma_3 / 2} = \begin{pmatrix} e^{i\varphi_3 / 2} & 0 \\ 0 & e^{-i\varphi_3 / 2} \end{pmatrix} = \cos \frac{\varphi_3}{2} \sigma_0 + i \sin \frac{\varphi_3}{2} \sigma_3 \quad (24)$$

где σ_0 - единичная матрица, а σ_3 третья матрица Паули. Пройдя по всей решетке, мы таким образом определим поток поля через всю плоскость xy . Графически преобразование можно представить следующим образом:

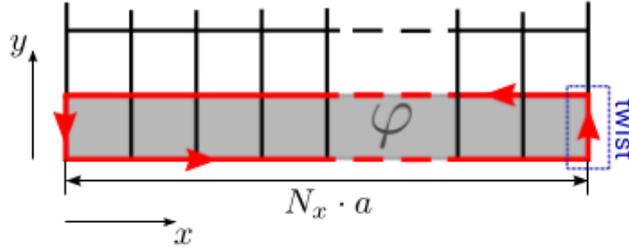


Рис. 3: Поток поля через плоскость xy

Такое преобразование можно представить в виде подкрученных граничных условий:

$$\begin{cases} U_y(N, n_y, n_z, n_t) = U_y(0, n_y, n_z, n_t) e^{i\phi} \\ U_\mu(N_x, n_y, n_z, n_t) = U_\mu(0, n_y, n_z, n_t) \quad \mu \neq y \\ U_\mu(n_x, N_y, n_z, n_t) = U_\mu(n_x, 0, n_z, n_t) \\ U_\mu(n_x, n_y, N_z, n_t) = U_\mu(n_x, n_y, 0, n_t) \\ U_\mu(n_x, n_y, n_z, N_t) = U_\mu(n_x, n_y, n_z, 0) \end{cases} \quad (25)$$

Применение данных граничных условий в коде программы приведено в приложении.

3.3 Хромомагнитное поле. Способ 2 (угловой)

Второй способ, описанный в статье [2], использует абелеву параметризацию калировочных полей U :

$$U_\mu(x) = \begin{pmatrix} \cos \phi_\mu(x) e^{i\theta_\mu(x)} & \sin \phi_\mu(x) e^{i\chi_\mu(x)} \\ -\sin \phi_\mu(x) e^{-i\chi_\mu(x)} & \cos \phi_\mu(x) e^{-i\theta_\mu(x)} \end{pmatrix} \quad (26)$$

где угловые переменные лежат в пределах $\phi \in [0, \pi/2]$, $\theta, \chi \in [-\pi, \pi]$. Влияние Абелева поля имеет место только на абелевой части линковой переменной, то есть на диагональных элементах. Тогда наиболее естественный способ ввести поле, предложен в статье т'Хофта [10]. Влияние поля φ , направленного вдоль оси z , на переменную U выглядит следующим образом:

$$U_\mu(x) = \begin{pmatrix} \cos \phi_\mu(x) e^{i(\theta_\mu(x) + \varphi_\mu(x))} & \sin \phi_\mu(x) e^{i\chi_\mu(x)} \\ -\sin \phi_\mu(x) e^{-i\chi_\mu(x)} & \cos \phi_\mu(x) e^{-i(\theta_\mu(x) + \varphi_\mu(x))} \end{pmatrix} \quad (27)$$

где $\varphi_\mu(x) = \varphi$, для линков на границе, то есть на месте (N_x, n_y, n_z, n_t) для $\mu = y$ и $\varphi_\mu(x) = 0$ для остальных переменных.

В терминах граничных условий данное преобразование примет вид:

$$U_\mu^0(x) = \begin{cases} U_\mu^0(z) \cos \varphi - U_\mu^3(x) \sin \varphi \\ \text{для } x = (N_x, n_y, n_z, n_t), \\ \mu = 2 \end{cases}, \quad U_\mu^3(x) = \begin{cases} U_\mu^0(z) \sin \varphi + U_\mu^3(x) \cos \varphi \\ \text{для } x = (N_x, n_y, n_z, n_t), \\ \mu = 2 \end{cases} \quad (28)$$

где U_μ^i соотносятся с обозначениями абелевой параметризации как:

$$U_\mu(x) = IU_\mu^0(x) + i\sigma_j U_\mu^j(x) = \begin{pmatrix} U_\mu^0(x) + iU_\mu^3(x) & U_\mu^2(x) + iU_\mu^1(x) \\ -U_\mu^2(x) + iU_\mu^1(x) & U_\mu^0(x) - iU_\mu^3(x) \end{pmatrix} \quad (29)$$

Применение данных граничных условий в коде программы приведено в приложении.

4 Характеристики вычислений

Все вычисления были проделаны на суперкомпьютере "ИТЭФ НИЦ "Курчатовский институт". Для вычисления натяжения струны была использована решетка размера 10^4 . Для поставленной задачи был написан код, а затем проанализированы результаты. Шаг решетки при $\beta = 3$ $a = 0.0576 \pm 0.0027 fm$. Вычисление шага решетки проводилось из соображений, что на решетке можно измерить безразмерную величину σa^2 , а затем найти a , зная из экспериментов натяжение струны $\sigma = (440 MeV)^2$.

5 Измеренные величины

Среднее значение плакета вычислялось по стандартной формуле

$$P = \sum_n \text{Tr} U_{\mu\nu}(n) \quad (30)$$

где $U_{\mu\nu}$ плакет и суммирование ведется по всем плакетам. Для наблюдения поля удобнее было рассматривать плакет в различных плоскостях отдельно. Для подсчета величины в одной плоскости следует зафиксировать величины $\mu\nu$. Для вычисления натяжения струны было использовано "Creutz ratio":

$$\chi(I, J) = -\log \frac{W(I, J)W(I-1, J-1)}{W(I-1, J)W(I, J-1)} \quad (31)$$

Где $W(I, J)$ - петля Вильсона размера $I * J$. "Creutz ratio" - способ вычисления натяжения струны. Вообще говоря, поведение петли может быть нетривиальным. В пределе сильной связи имеет место закон площадей: $\langle W \rangle \sim e^{-\sigma IJ}$, однако при уменьшении связи и стремлении к непрерывному пределу более существенный вклад дает периметрическая зависимость. Поэтому удобно рассматривать величину χ , в которой вклады от периметров сокращаются. Величина χ при больших β стремиться к натяжению струны, заданной формулой

$$\sigma a^2(g) = \frac{\sigma}{\Lambda^2} \exp\left(-\frac{1}{\beta_0 g^2}\right) (\beta_0 g)^{-\frac{\beta_1}{\beta_0^2}} (1 + O(g^2)) \quad (32)$$

при достаточно больших петлях. Формула (32) получается из теории ренорм-группы и связывает голый параметр связи g с размером шага решетки a . Для того, чтобы понять, какие размеры петель необходимо брать, был построен график зависимости χ от площади петли и оценены размеры петель, на которых получаются достоверные результаты. Натяжение струны нужно искать при β больше, чем 2.3 (критическое значение β), однако ошибка при расчете χ сильно растёт с увеличением площади петли. Для измерений были выбраны наиболее оптимальные значения β , близкие к 3.

6 Полученные результаты

Влияние внешнего абелева (хромо)магнитного поля было изучено для $U(1)$ и $SU(2)$ калибровочных теорий. В $U(1)$ теории влияние внешнего поля более наглядно и очевидно. Как оказалось, существуют различия в поведении системы при добавлении внешнего поля для этих двух калибровочных теорий. Рассмотрим результаты чуть подробнее.

6.1 $U(1)$ теория

Для $U(1)$ теории поле было введено способом 1. Домножение на матрицу было заменено на домножение на комплексное число $\exp i\varphi$. В предельном случае $\beta \gg 1$, аналогично $SU(2)$ теории ожидается сведение решетки к виду:

$$\begin{aligned} U_1^{ext} &= U_4^{ext} = U_3^{ext} = 1 \\ U_2^{ext} &= \cos ax_1 H + i \sin ax_1 H \end{aligned} \quad (33)$$

А плакет стремится к значению $\cos aH$. Было замечено, что в этом пределе ($\beta \gg 1$), для $U(1)$ теории плакет падает только в плоскости xu , как и ожидалось. Результаты измерений можно увидеть на Рис. 4.

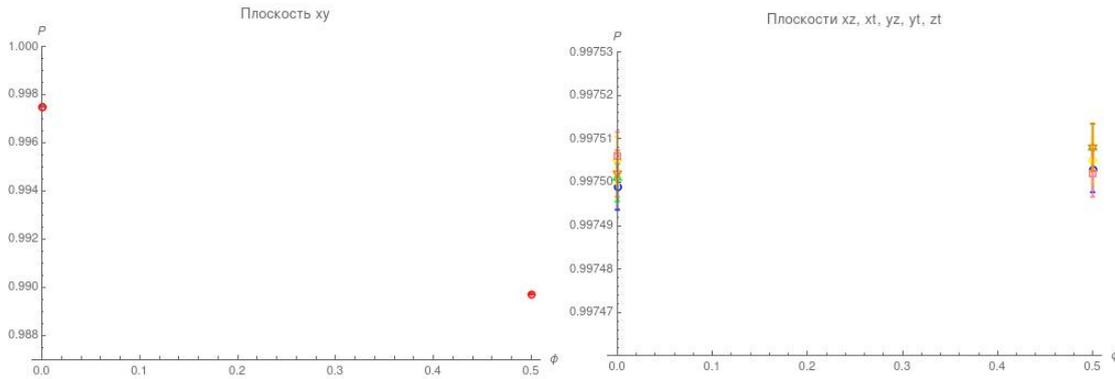


Рис. 4: Зависимость плакета от поля для группы $U(1)$ для решетки 10^4

Такое поведение согласуется с аналитическими предсказаниями о поведении плакета во внешнем поле.

6.2 $SU(2)$ теория

Для $SU(2)$ теории получились немного другие результаты.

Чтобы удостовериться, что влияние поля имеет место в пределе больших β , плакет был измерен для $\beta \gg 1$. Такой предел "выключает" динамическое поле на решетке и можно наблюдать влияние внешнего приложенного абелева поля.

Поле было введено двумя различными способами, матричным и угловым. Результаты измерений можно увидеть на Рис. 5

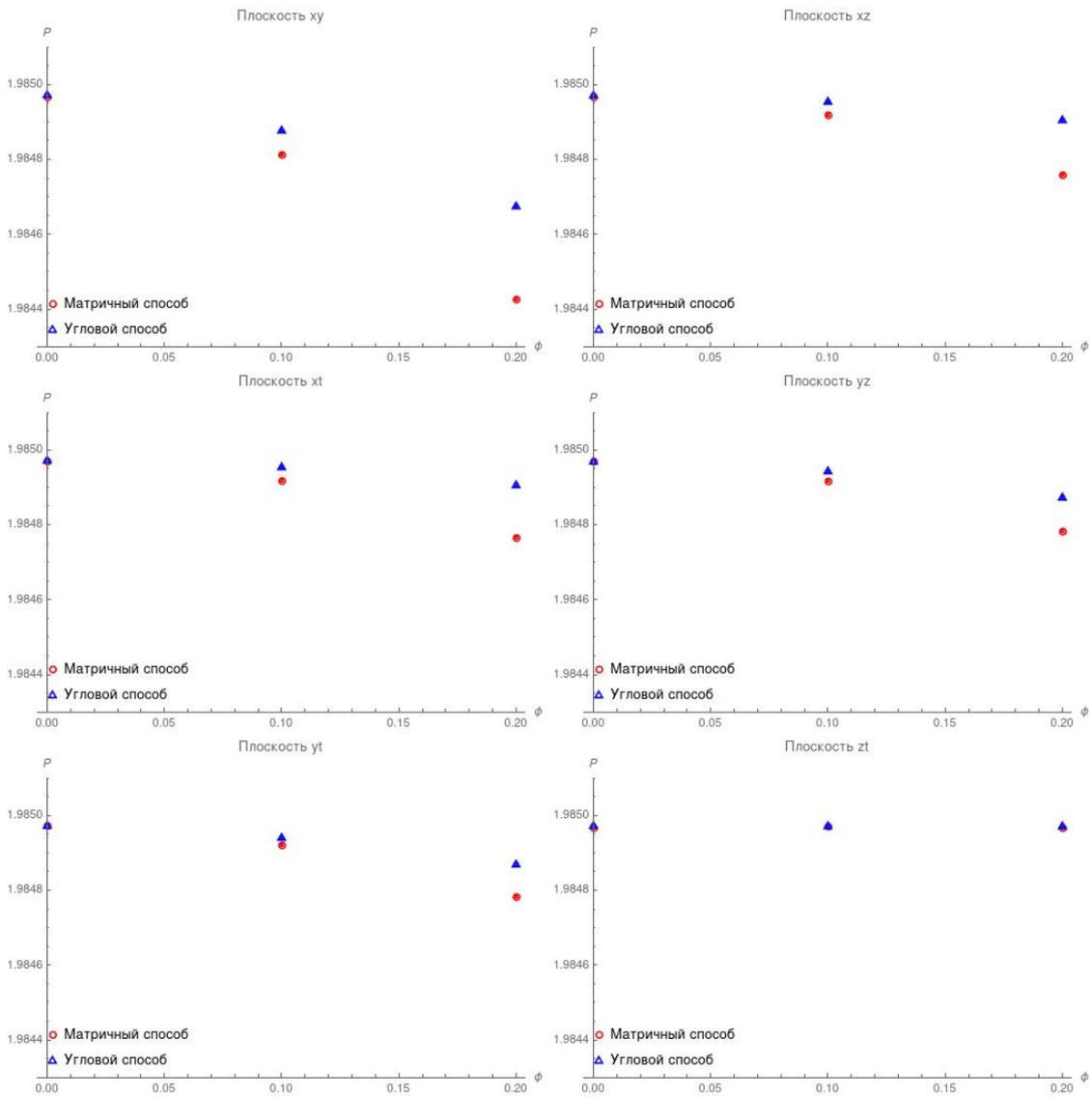


Рис. 5: Зависимость плакета от поля, введенного двумя различными способами на решетке 10^4

На графиках можно заметить, что в обоих способах, существенное влияние имеет место только в плоскости $xу$, как и ожидалось, однако утверждать, что оба способа приводят к одному и тому же результату нельзя.

Далее все вычисления сделаны для матричного способа.

Измеряя плакет на малых полях, из предыдущих графиков видно, что поле влияет не только на плоскость $xу$, но и на другие: xz , xt , yz , yt , zt . Анализ поведения зависимости плакета от поля показал, что в плоскости $xу$ зависимость имеет квадратичный характер, в то время как в других плоскостях, квадратичной зависимостью можно пренебречь. Таким образом, подобного рода зависимость дает возможность брать поля такие, что в плоскости $xу$ влияние поля будет существенным, в то время как в остальных плоскостях, малым. На рисунке 8 проиллюстрирована зависимость плакета от поля в плоскостях $xу$ и xz .

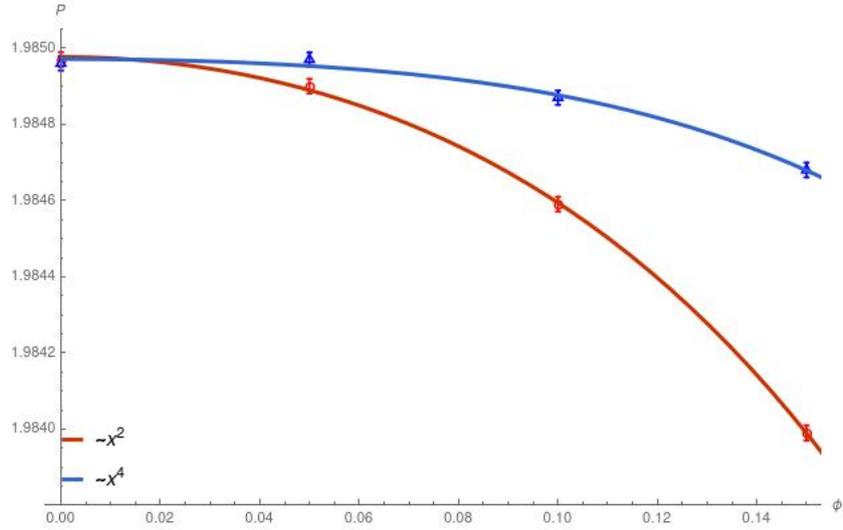


Рис. 6: Зависимость плакета от поля для $SU(2)$ теории при $\beta = 100$ на 2000 конфигурациях

Таким образом, влияние поля на неабелеву калибровочную теорию имеет нетривиальный характер, в отличие от абелевой.

Расчеты плакета были проведены для $\beta = 3$. Оказалось, что для β порядка единицы, чтобы увидеть влияние поля, требуется статистика в 100 раз больше, чем для $\beta = 100$. Увеличение статистики в сто раз дало результаты.

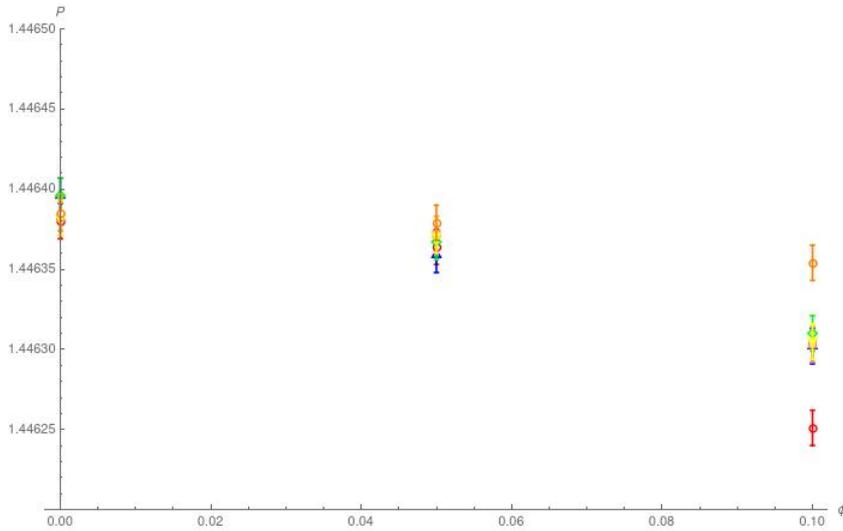


Рис. 7: Зависимость плакета от поля для $SU(2)$ теории при $\beta = 3$ на 200000 конфигурациях

Для того чтобы измерить натяжение струны и использовать формулу (31) необходимо выбрать оптимальные параметры измерения. Было зафиксировано $\beta = 3$, при этом шаг решетки имеет значение $a = 0.0576 \pm 0.0027 fm$. Так, чтобы использовать формулу (31) необходимо определить с какого размера петли, полученные значения будут выходить на плато и соответствовать слабой связи. На Рис.8 видно, что χ выходит на плато при площади петли больше 16, это значит, что для измерения натяжения струны, следует считать петли размера больше 16.

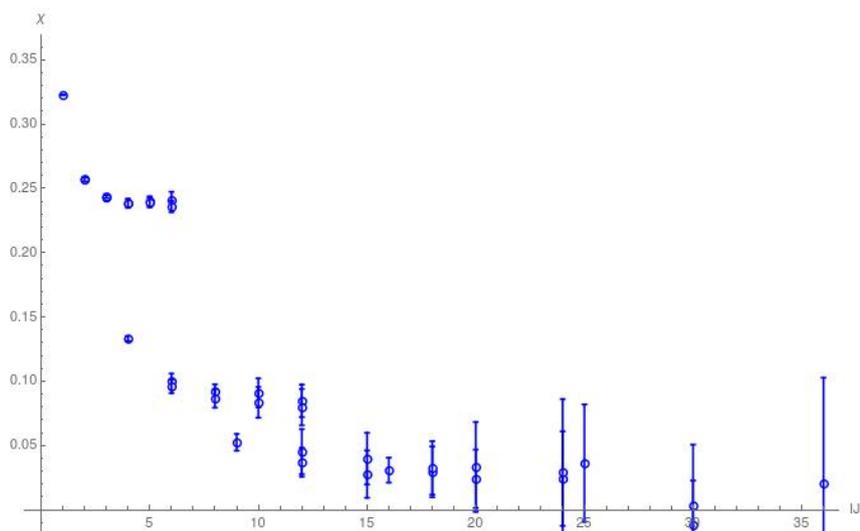


Рис. 8: Зависимость χ от площади петли

Такое слабое влияние поля на плакет ставит под сомнение возможность расчета натяжения струны с необходимой точностью. Предварительные данные изображены на Рис.9 для плоскости $x\mu$. В пределах ошибок зависимости натяжения струны от φ не видно. В перспективе возможно применение других алгоритмов для подсчета петель, уменьшающих ошибку.

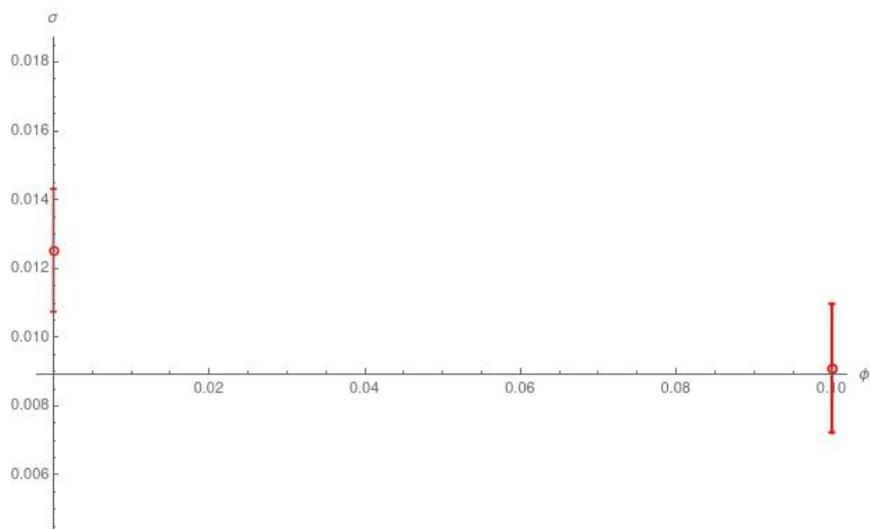


Рис. 9: Зависимость σ от приложенного поля φ на 2000 конфигурациях

7 Заключение

Решеточное моделирование дает большой простор для исследования физики высоких энергий. Существует много открытых вопросов в этой области и направлений исследования. В частности интересны новые эффекты, возникающие в КХД в различных условиях.

В данной работе мы исследовали глюодинамику во внешнем абелевом хромоманнитном поле. Влияние поля было исследовано двумя различными способами. Несмотря на то, что, казалось бы, оба способа должны давать одинаковые результаты, существуют различия, которые не дают нам сделать вывод, что оба способа ввести граничные условия эквивалентны. Это дает почву для дальнейших исследований двух методов введения поля и определения наиболее физически осмысленного подхода.

Матричный способ был исследован более подробно. Для него удалось выделить зависимость плакета от поля для предельного случая, однако для малых β вычисление плакета представляет собой достаточно долгое вычисление, так как для получения явного результата необходимо большое количество конфигураций. Вследствие чего, предварительные результаты, полученные для натяжения струны, говорят нам о том, что в пределах ошибок влияние поля не наблюдается. Применение других, менее тривиальных алгоритмов для расчета петель может помочь уменьшить ошибки в расчетах. В перспективе планируется опробовать новые методы оценки ошибок и получить наглядный результат для натяжения струны.

Список литературы

- [1] Heng-Tong Ding, Frithjof Karsch и Swagato Mukherjee. *Thermodynamics of strong-interaction matter from Lattice QCD*. 2015. arXiv: 1504.05274 [hep-lat].
- [2] VI Demchik и VV Skalozub. «On the spontaneous creation of chromomagnetic fields at high temperature». в: *Physics of Atomic Nuclei* 71.1 (2008), с. 180—186.
- [3] Paolo Cea, Leonardo Cosmai и Massimo D’Elia. «QCD dynamics in a constant chromomagnetic field». в: *Journal of High Energy Physics* 2007.12 (дек. 2007), с. 097—097. ISSN: 1029-8479. DOI: 10.1088/1126-6708/2007/12/097. URL: <http://dx.doi.org/10.1088/1126-6708/2007/12/097>.
- [4] Paolo Cea и Leonardo Cosmai. «Color dynamics in external fields». в: *Journal of High Energy Physics* 2005.08 (2005), с. 079.
- [5] Tuomas Lappi и L McLerran. «Some features of the glasma». в: *Nuclear Physics A* 772.3-4 (2006), с. 200—212.
- [6] Leonardo Campanelli и Marco Ruggieri. «Probing the QCD vacuum with an Abelian chromomagnetic field: A study within an effective model». в: *Physical Review D* 80.3 (2009), с. 034014.
- [7] Michael Creutz. *Quarks, gluons and lattices*. т. 8. Cambridge University Press, 1983.
- [8] Christof Gattringer и Christian Lang. *Quantum chromodynamics on the lattice: an introductory presentation*. т. 788. Springer Science & Business Media, 2009.
- [9] Natalia V Kolomojets и Vladimir V Skalozub. «Color structure of gluon field magnetic mass». в: *International Journal of Modern Physics A* 34.09 (2019), с. 1950052.
- [10] G t t Hooft. «A property of electric and magnetic flux in non-Abelian gauge theories». в: *Nuclear physics: B* 153 (1979), с. 141—160.

Приложение

В данном приложении часть кода для алгоритма heat bath, вычисляющая коэффициент k , получающийся следующим образом:

$$k = \left| \sum_{\alpha=1}^6 U_{\alpha} \right|^{\frac{1}{2}} \quad (34)$$

где U_{α} скобки, в которые входит рассчитывающийся линк, образуя плакет. Именно эта часть кода имеет значение для введения поля, потому что при расчете линков на границе требуется подкрутка фазы. В коде представлено сразу 2 способа, угловой способ в данной части кода закоментирована, чтобы перейти от матричного к угловому следует разкомментировать подсчет скобки соответствующий матричному способу и закоментировать другой. Функция Phase() возвращает элемент группы $SU(2)$, измененный по правилу (28)

```
void kcounter_new (SuperArrayDim4<Link<MatrixSU2>> & arr, int i, int j, int k, int m, int l, double & coef, MatrixSU2 & Matt){
MatrixSU2 Res(0);
vector<MatrixSU2> M(6);
Matrix matrix;
MatrixSU2 phase;
double f[4] = {cos(phi), 0, 0, sin(phi)};
phase.set_matrix(f);
//Если я на границе плоскости xy, то следует отдельно выделить два случая, i = n-1 и i = 0
switch(i){
case n - 1:
switch(l){
case 0:
// здесь введено поле для способа 1
M[0] = arr[i + 1][j][k][m].get(1)* phase *arr[i][j+ 1][k][m].get(0).Inverse()*arr[i][j][k][m].get(1).Inverse();
// здесь введено поле для способа 2
//M[0] = arr[i + 1][j][k][m].get(1).Phase() *arr[i][j + 1][k][m].get(0).Inverse()*arr[i][j][k][m].get(1).Inverse();
M[1] = arr[i + 1][j][k][m].get(2) * arr[i][j][k + 1][m].get(0).Inverse()*arr[i][j][k][m].get(2).Inverse();
M[2] = arr[i + 1][j][k][m].get(3) * arr[i][j][k][m + 1].get(0).Inverse()*arr[i][j][k][m].get(3).Inverse();
// здесь введено поле для способа 1
M[3] = phase.Inverse() * arr[i + 1][j - 1][k][m].get(1).Inverse() * arr[i][j - 1][k][m].get(0).Inverse() * arr[i][j - 1][k][m].get(1);
// здесь введено поле для способа 2
// M[3] = (arr[i + 1][j - 1][k][m].get(1).Phase()).Inverse() * arr[i][j - 1][k][m].get(0).Inverse() * arr[i][j - 1][k][m].get(1);
M[4] = arr[i + 1][j][k - 1][m].get(2).Inverse() * arr[i][j][k - 1][m].get(0).Inverse() * arr[i][j][k - 1][m].get(2);
M[5] = arr[i + 1][j][k][m - 1].get(3).Inverse() * arr[i][j][k][m - 1].get(0).Inverse() * arr[i][j][k][m - 1].get(3);
break;
case 1:
// здесь введено поле для способа 1
M[0] = arr[i][j + 1][k][m].get(0)* phase.Inverse() * arr[i + 1][j][k][m].get(1).Inverse() * arr[i][j][k][m].get(0).Inverse();
// здесь введено поле для способа 2
// M[0] = arr[i][j + 1][k][m].get(0)*(arr[i + 1][j][k][m].get(1).Phase()).Inverse() * arr[i][j][k][m].get(0).Inverse();
M[1] = arr[i][j + 1][k][m].get(2)*arr[i][j][k + 1][m].get(1).Inverse()*arr[i][j][k][m].get(2).Inverse();
M[2] = arr[i][j + 1][k][m].get(3)*arr[i][j][k][m + 1].get(1).Inverse()*arr[i][j][k][m].get(3).Inverse();
M[3] = arr[i - 1][j + 1][k][m].get(0).Inverse() * arr[i - 1][j][k][m].get(1).Inverse() * arr[i - 1][j][k][m].get(0);
M[4] = arr[i][j + 1][k - 1][m].get(2).Inverse() * arr[i][j][k - 1][m].get(1).Inverse() * arr[i][j][k - 1][m].get(2);
M[5] = arr[i][j + 1][k][m - 1].get(3).Inverse() * arr[i][j][k][m - 1].get(1).Inverse() * arr[i][j][k][m - 1].get(3);
break;
case 2:
M[0] = arr[i][j][k + 1][m].get(0)*arr[i + 1][j][k][m].get(2).Inverse()*arr[i][j][k][m].get(0).Inverse();
M[1] = arr[i][j][k + 1][m].get(1)*arr[i][j + 1][k][m].get(2).Inverse()*arr[i][j][k][m].get(1).Inverse();
M[2] = arr[i][j][k + 1][m].get(3)*arr[i][j][k][m + 1].get(2).Inverse()*arr[i][j][k][m].get(3).Inverse();
M[3] = arr[i - 1][j][k + 1][m].get(0).Inverse() * arr[i - 1][j][k][m].get(2).Inverse() * arr[i - 1][j][k][m].get(0);
M[4] = arr[i][j - 1][k + 1][m].get(1).Inverse() * arr[i][j - 1][k][m].get(2).Inverse() * arr[i][j - 1][k][m].get(1);
M[5] = arr[i][j][k + 1][m - 1].get(3).Inverse() * arr[i][j][k][m - 1].get(2).Inverse() * arr[i][j][k][m - 1].get(3);
break;
case 3:
M[0] = arr[i][j][k][m + 1].get(0)*arr[i + 1][j][k][m].get(3).Inverse()*arr[i][j][k][m].get(0).Inverse();
M[1] = arr[i][j][k][m + 1].get(1)*arr[i][j + 1][k][m].get(3).Inverse()*arr[i][j][k][m].get(1).Inverse();
M[2] = arr[i][j][k][m + 1].get(2)*arr[i][j][k + 1][m].get(3).Inverse()*arr[i][j][k][m].get(2).Inverse();
M[3] = arr[i - 1][j][k][m + 1].get(0).Inverse() * arr[i - 1][j][k][m].get(3).Inverse() * arr[i - 1][j][k][m].get(0);
M[4] = arr[i][j - 1][k][m + 1].get(1).Inverse() * arr[i][j - 1][k][m].get(3).Inverse() * arr[i][j - 1][k][m].get(1);
M[5] = arr[i][j][k - 1][m + 1].get(2).Inverse() * arr[i][j][k - 1][m].get(3).Inverse() * arr[i][j][k - 1][m].get(2);
break;
}
```

```

}

for(auto x: M){
    Res += x;
}
Res.groupElement(matrix);
break;

case 0:
switch(1){
case 0:
M[0] = arr[i + 1][j][k][m].get(1) * arr[i][j + 1][k][m].get(0).Inverse() * arr[i][j][k][m].get(1).Inverse();
M[1] = arr[i + 1][j][k][m].get(2) * arr[i][j][k + 1][m].get(0).Inverse() * arr[i][j][k][m].get(2).Inverse();
M[2] = arr[i + 1][j][k][m].get(3) * arr[i][j][k][m + 1].get(0).Inverse() * arr[i][j][k][m].get(3).Inverse();
M[3] = arr[i + 1][j - 1][k][m].get(1).Inverse() * arr[i][j - 1][k][m].get(0).Inverse() * arr[i][j - 1][k][m].get(1);
M[4] = arr[i + 1][j][k - 1][m].get(2).Inverse() * arr[i][j][k - 1][m].get(0).Inverse() * arr[i][j][k - 1][m].get(2);
M[5] = arr[i + 1][j][k][m - 1].get(3).Inverse() * arr[i][j][k][m - 1].get(0).Inverse() * arr[i][j][k][m - 1].get(3);
break;
case 1:
M[0] = arr[i][j + 1][k][m].get(0) * arr[i + 1][j][k][m].get(1).Inverse() * arr[i][j][k][m].get(0).Inverse();
M[1] = arr[i][j + 1][k][m].get(2) * arr[i][j][k + 1][m].get(1).Inverse() * arr[i][j][k][m].get(2).Inverse();
M[2] = arr[i][j + 1][k][m].get(3) * arr[i][j][k][m + 1].get(1).Inverse() * arr[i][j][k][m].get(3).Inverse();
// здесь введено поле для способа 1
M[3] = phase * arr[i - 1][j + 1][k][m].get(0).Inverse() * arr[i - 1][j][k][m].get(1).Inverse() * arr[i - 1][j][k][m].get(0);
// здесь введено поле для способа 2
//M[3] = (arr[i][j][k][l].get(0) * arr[i - 1][j + 1][k][m].get(0).Inverse() * arr[i - 1][j][k][m].get(1).Inverse()).Phase();
M[4] = arr[i][j + 1][k - 1][m].get(2).Inverse() * arr[i][j][k - 1][m].get(1).Inverse() * arr[i][j][k - 1][m].get(2);
M[5] = arr[i][j + 1][k][m - 1].get(3).Inverse() * arr[i][j][k][m - 1].get(1).Inverse() * arr[i][j][k][m - 1].get(3);
break;
case 2:
M[0] = arr[i][j][k + 1][m].get(0) * arr[i + 1][j][k][m].get(2).Inverse() * arr[i][j][k][m].get(0).Inverse();
M[1] = arr[i][j][k + 1][m].get(1) * arr[i][j + 1][k][m].get(2).Inverse() * arr[i][j][k][m].get(1).Inverse();
M[2] = arr[i][j][k + 1][m].get(2) * arr[i][j][k][m + 1].get(2).Inverse() * arr[i][j][k][m].get(2).Inverse();
M[3] = arr[i - 1][j][k + 1][m].get(0).Inverse() * arr[i - 1][j][k][m].get(2).Inverse() * arr[i - 1][j][k][m].get(0);
M[4] = arr[i][j - 1][k + 1][m].get(1).Inverse() * arr[i][j - 1][k][m].get(2).Inverse() * arr[i][j - 1][k][m].get(1);
M[5] = arr[i][j][k + 1][m - 1].get(3).Inverse() * arr[i][j][k][m - 1].get(2).Inverse() * arr[i][j][k][m - 1].get(3);
break;
case 3:
M[0] = arr[i][j][k][m + 1].get(0) * arr[i + 1][j][k][m].get(3).Inverse() * arr[i][j][k][m].get(0).Inverse();
M[1] = arr[i][j][k][m + 1].get(1) * arr[i][j + 1][k][m].get(3).Inverse() * arr[i][j][k][m].get(1).Inverse();
M[2] = arr[i][j][k][m + 1].get(2) * arr[i][j][k + 1][m].get(3).Inverse() * arr[i][j][k][m].get(2).Inverse();
M[3] = arr[i - 1][j][k][m + 1].get(0).Inverse() * arr[i - 1][j][k][m].get(3).Inverse() * arr[i - 1][j][k][m].get(0);
M[4] = arr[i][j - 1][k][m + 1].get(1).Inverse() * arr[i][j - 1][k][m].get(3).Inverse() * arr[i][j - 1][k][m].get(1);
M[5] = arr[i][j][k - 1][m + 1].get(2).Inverse() * arr[i][j][k - 1][m].get(3).Inverse() * arr[i][j][k - 1][m].get(2);
break;
}

for(auto x: M){
    Res += x;
}
Res.groupElement(matrix);
break;

default:
switch(1){
case 0:
M[0] = arr[i + 1][j][k][m].get(1) * arr[i][j + 1][k][m].get(0).Inverse() * arr[i][j][k][m].get(1).Inverse(); // здесь поле
M[1] = arr[i + 1][j][k][m].get(2) * arr[i][j][k + 1][m].get(0).Inverse() * arr[i][j][k][m].get(2).Inverse();
M[2] = arr[i + 1][j][k][m].get(3) * arr[i][j][k][m + 1].get(0).Inverse() * arr[i][j][k][m].get(3).Inverse();
M[3] = arr[i + 1][j - 1][k][m].get(1).Inverse() * arr[i][j - 1][k][m].get(0).Inverse() * arr[i][j - 1][k][m].get(1); // здесь поле
M[4] = arr[i + 1][j][k - 1][m].get(2).Inverse() * arr[i][j][k - 1][m].get(0).Inverse() * arr[i][j][k - 1][m].get(2);
M[5] = arr[i + 1][j][k][m - 1].get(3).Inverse() * arr[i][j][k][m - 1].get(0).Inverse() * arr[i][j][k][m - 1].get(3);
break;
case 1:
M[0] = arr[i][j + 1][k][m].get(0) * arr[i + 1][j][k][m].get(1).Inverse() * arr[i][j][k][m].get(0).Inverse(); // здесь поле
M[1] = arr[i][j + 1][k][m].get(2) * arr[i][j][k + 1][m].get(1).Inverse() * arr[i][j][k][m].get(2).Inverse();
M[2] = arr[i][j + 1][k][m].get(3) * arr[i][j][k][m + 1].get(1).Inverse() * arr[i][j][k][m].get(3).Inverse();
M[3] = arr[i - 1][j + 1][k][m].get(0).Inverse() * arr[i - 1][j][k][m].get(1).Inverse() * arr[i - 1][j][k][m].get(0);
M[4] = arr[i][j + 1][k - 1][m].get(2).Inverse() * arr[i][j][k - 1][m].get(1).Inverse() * arr[i][j][k - 1][m].get(2);
M[5] = arr[i][j + 1][k][m - 1].get(3).Inverse() * arr[i][j][k][m - 1].get(1).Inverse() * arr[i][j][k][m - 1].get(3);
break;
case 2:
M[0] = arr[i][j][k + 1][m].get(0) * arr[i + 1][j][k][m].get(2).Inverse() * arr[i][j][k][m].get(0).Inverse();
M[1] = arr[i][j][k + 1][m].get(1) * arr[i][j + 1][k][m].get(2).Inverse() * arr[i][j][k][m].get(1).Inverse();
M[2] = arr[i][j][k + 1][m].get(3) * arr[i][j][k][m + 1].get(2).Inverse() * arr[i][j][k][m].get(3).Inverse();
M[3] = arr[i - 1][j][k + 1][m].get(0).Inverse() * arr[i - 1][j][k][m].get(2).Inverse() * arr[i - 1][j][k][m].get(0);
M[4] = arr[i][j - 1][k + 1][m].get(1).Inverse() * arr[i][j - 1][k][m].get(2).Inverse() * arr[i][j - 1][k][m].get(1);
M[5] = arr[i][j][k + 1][m - 1].get(3).Inverse() * arr[i][j][k][m - 1].get(2).Inverse() * arr[i][j][k][m - 1].get(3);
break;
case 3:

```

```

M[0] = arr[i][j][k][m + 1].get(0) * arr[i + 1][j][k][m].get(3).Inverse() * arr[i][j][k][m].get(0).Inverse();
M[1] = arr[i][j][k][m + 1].get(1) * arr[i][j + 1][k][m].get(3).Inverse() * arr[i][j][k][m].get(1).Inverse();
M[2] = arr[i][j][k][m + 1].get(2) * arr[i][j][k + 1][m].get(3).Inverse() * arr[i][j][k][m].get(2).Inverse();
M[3] = arr[i - 1][j][k][m + 1].get(0).Inverse() * arr[i - 1][j][k][m].get(3).Inverse() * arr[i - 1][j][k][m].get(0);
M[4] = arr[i][j - 1][k][m + 1].get(1).Inverse() * arr[i][j - 1][k][m].get(3).Inverse() * arr[i][j - 1][k][m].get(1);
M[5] = arr[i][j][k - 1][m + 1].get(2).Inverse() * arr[i][j][k - 1][m].get(3).Inverse() * arr[i][j][k - 1][m].get(2);
break;
}
for(auto x: M){
    Res += x;
}
Res.groupElement(matrix);
break;
}
complex<double>coeff = sqrt(matrix[0][0]*matrix[1][1] - matrix[0][1]*matrix[1][0]);
coef = coeff.real();
Matt = Res / coef;
}

```