



ФИЗИЧЕСКИЙ  
ФАКУЛЬТЕТ  
МГУ ИМЕНИ  
М.В. ЛОМОНОСОВА

*teach-in*  
ЛЕКЦИИ УЧЕНЫХ МГУ

# КОМПЬЮТЕРНО-АЛГЕБРАИЧЕСКИЕ МЕТОДЫ ПЕРТУРБАТИВНЫХ ВЫЧИСЛЕНИЙ В КВАНТОВОЙ ТЕОРИИ ПОЛЯ

ШИРОКОВ  
ИЛЬЯ ЕВГЕНЬЕВИЧ

ФИЗФАК МГУ

КОНСПЕКТ ПОДГОТОВЛЕН  
СТУДЕНТАМИ, НЕ ПРОХОДИЛ  
ПРОФ. РЕДАКТУРУ И МОЖЕТ  
СОДЕРЖАТЬ ОШИБКИ.  
СЛЕДИТЕ ЗА ОБНОВЛЕНИЯМИ  
НА [VK.COM/TEACHINMSU](https://vk.com/teachinmsu).

ЕСЛИ ВЫ ОБНАРУЖИЛИ  
ОШИБКИ ИЛИ ОПЕЧАТКИ,  
ТО СООБЩИТЕ ОБ ЭТОМ,  
НАПИСАВ СООБЩЕСТВУ  
[VK.COM/TEACHINMSU](https://vk.com/teachinmsu).

# Содержание

<b>1</b>	<b>Лекция 1. История развития и современное состояние методов компьютерной алгебры</b>	<b>4</b>
1.1	Вычисление аномального магнитного момента электрона . . . . .	4
1.2	Проблема вычисления следов $\gamma$ -матриц . . . . .	5
1.3	История развития методов компьютерной алгебры . . . . .	5
1.4	Философское отступление. Вопрос о «дегуманизации науки» . . . . .	7
1.5	План Курса . . . . .	8
<b>2</b>	<b>Лекция 2. Методы генерации диаграмм. Основы теории графов</b>	<b>9</b>
2.1	Диаграммы Фейнмана. Производящий функционал . . . . .	9
2.1.1	Вычисление симметричных коэффициентов диаграмм . . . . .	10
2.2	Существующие реализации генерации диаграмм. Программа <i>FeynArts</i> .	11
2.2.1	Пример генерации топологии в программе <i>FeynArts</i> . . . . .	12
2.3	Проблема изоморфизма графов . . . . .	13
2.4	Теория графов. Генерация диаграмм в программе <i>Qgraf</i> . . . . .	14
2.4.1	Способ задания графа при помощи матрицы смежности . . . . .	15
<b>3</b>	<b>Лекция 3. Демонстрация работы в программе <i>Qgraf</i></b>	<b>18</b>
3.1	Установка программы <i>Qgraf</i> . . . . .	18
3.2	Процесс работы в программе <i>Qgraf</i> . . . . .	18
3.3	Различные примеры генерации графов в программе <i>Qgraf</i> . . . . .	23
<b>4</b>	<b>Лекция 4. Демонстрация работы в программе <i>FeynArts</i></b>	<b>28</b>
4.1	Установка пакета <i>FeynArts</i> . . . . .	28
4.2	Работа с пакетом <i>FeynArts</i> . . . . .	28
4.3	Расчет амплитуд процессов в программе <i>FeynArts</i> . . . . .	32
4.4	Ограничения пакета <i>FeynArts</i> . . . . .	32
<b>5</b>	<b>Лекция 5. Представления петлевых интегралов Фейнмана</b>	<b>33</b>
5.1	Общий подход к рассмотрению диаграмм . . . . .	33
5.1.1	Простейший пример вычисления петлевых интегралов . . . . .	33
5.2	Примеры интегралов Фейнмана . . . . .	34
5.3	Представления интегралов . . . . .	36
5.3.1	Представление Швингера . . . . .	36
5.3.2	Представление Фейнмана . . . . .	37
5.4	Пример вычисления петлевого интеграла с использованием различных представлений . . . . .	38

<b>6</b>	<b>Лекция 6. IBP reduction. Интегрирование по частям интегралов Фейнмана</b>	<b>40</b>
6.1	Тензорные структуры в интегралах . . . . .	40
6.2	Метод интегрирования по частям . . . . .	41
6.3	Использование Лоренц-инвариантности . . . . .	44
6.4	Безмасштабные интегралы (Scaleless) . . . . .	45
6.4.1	Угловой интеграл . . . . .	45
<b>7</b>	<b>Лекция 7. Безмасштабные интегралы. Критерий Ли. Алгоритм Лапорта</b>	<b>47</b>
7.1	Идея метода IBP reduction . . . . .	47
7.2	Безмасштабные интегралы . . . . .	48
7.3	Представление Ли-Померанского . . . . .	49
7.4	Критерий Ли . . . . .	50
7.4.1	Пример применения критерия Ли . . . . .	51
7.5	Алгоритм Лапорта . . . . .	51
<b>8</b>	<b>Лекция 8. Программы LiteRed и Kira</b>	<b>53</b>
8.1	Программа LiteRed . . . . .	53
8.1.1	Установка программы LiteRed . . . . .	53
8.1.2	Пример работы в программе LiteRed . . . . .	53
8.1.3	Дополнительные замечания по работе с программой LiteRed . . . . .	55
8.2	Программа Kira . . . . .	55
8.2.1	Установка программы Kira . . . . .	56
8.2.2	Работа в программе Kira . . . . .	56
<b>9</b>	<b>Лекция 9. Численное вычисление интегралов Фейнмана. Метод Меллина-Барнса</b>	<b>58</b>
9.1	Метод Меллина-Барнса . . . . .	58
9.1.1	Преобразование Меллина . . . . .	58
9.1.2	Обобщение формулы Меллина-Барнса . . . . .	60
9.2	Применение метода Меллина-Барнса к вычислению интегралов Фейнмана . . . . .	62
9.3	Леммы Барнса . . . . .	63
9.4	Связь Гамма-функций с гипергеометрическими функциями . . . . .	64
<b>10</b>	<b>Лекция 11. Численные методы вычисления интегралов Фейнмана. Метод разложения по секторам</b>	<b>65</b>

10.1	Метод разложения по секторам . . . . .	65
10.2	Проблемы метода разложения по секторам . . . . .	69
<b>11</b>	<b>Лекция 12. Программы AMBRE и FIESTA 4</b>	<b>70</b>
11.1	Пакеты AMBRE и MB . . . . .	70
11.1.1	Установка пакета AMBRE и MB . . . . .	70
11.2	Работа в программе AMBRE . . . . .	70
11.3	Программа FIESTA 4 . . . . .	72
11.4	Установка пакета FIESTA 4 . . . . .	72
11.5	Работа в программе FIESTA 4 . . . . .	72
<b>12</b>	<b>Лекция 13. Особенности вычисления диаграмм в неабелевых калибровочных теориях</b>	<b>75</b>
12.1	Диаграммы в неабелевых калибровочных теориях . . . . .	75
12.2	Свойства генераторов калибровочной группы . . . . .	76
12.3	Некоторые характерные конструкции, возникающие в неабелевых теориях . . . . .	80

# Лекция 1. История развития и современное состояние методов компьютерной алгебры

Курс посвящен компьютерным методам в теоретической физике. Для успешного усвоения материала курса необходимо иметь представление о сути и методах квантовой теории поля (далее - КТП).

Чаще всего аналитические вычисления методами КТП ведутся в низких порядках теории возмущений (древесное/однопетлевое приближения). Однако, опыт показывает, что вычисления в высоких порядках также представляют большой интерес.

## Вычисление аномального магнитного момента электрона

Одно из наиболее известных вычислений КТП — *вычисление аномального магнитного момента электрона*.

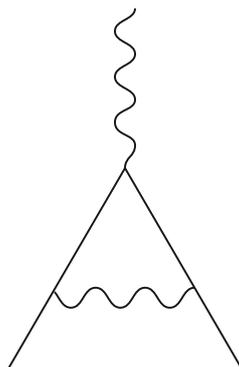


Рис. 1.1. Однопетлевая диаграмма для расчета аномального магнитного момента электрона

Так, если рассмотреть однопетлевую диаграмму (см. рис 1.1) эта диаграмма пропорциональна

$$\sim \gamma^\mu F_1(q) A_\mu + \sigma^{\mu\nu} \partial_\nu A_\mu F_2(q). \quad (1.1)$$

В последнем выражении форм-фактор:

$$F_2 = a_e \frac{g-2}{2}, \quad (1.2)$$

где  $a_e$  - аномальный магнитный момент электрона.

Аномальный магнитный момент вычислен в высоких порядках теории возмущений — вплоть до 5 петель.

В одной петле существует только одна диаграмма, представленная выше, которая дает вклад, который пропорционален  $\sim e^2$ .

В двухпетлевом приближении диаграмм будет значительно больше — 7 штук (см. рис 1.2). Их вклад будет пропорционален  $\sim e^4$ .

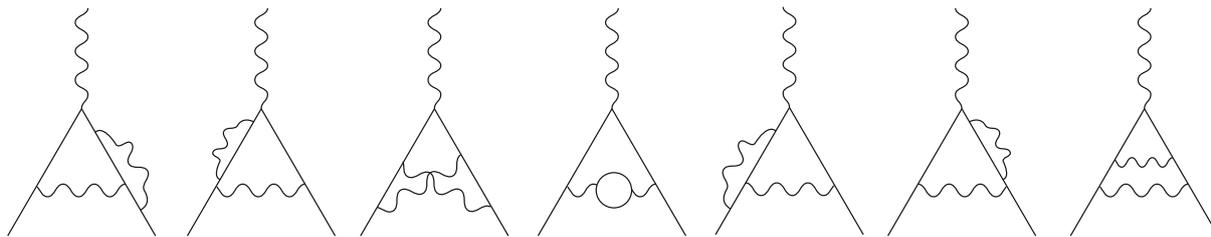


Рис. 1.2. Двухпетлевые диаграммы для вычисления аномального магнитного момента электрона

- В третьем порядке  $\sim e^6$  - 72 штуки.
- В четвертом  $\sim e^8$  - 891 штука
- В пятом  $\sim e^{10}$  - около 12000 штук (вычислены не все).

С каждым порядком количество диаграмм увеличивается примерно на порядок. Кроме того, увеличивается и сложность вычисления каждой диаграммы. Проблема вычисления *аномального магнитного момента* требует многопетлевого вычисления, что стимулировало развитие компьютерных методов.

## Проблема вычисления следов $\gamma$ -матриц

Как хорошо известно

$$\text{tr}(\gamma^\mu \gamma^\nu) = 4\eta^{\mu\nu}. \quad (1.3)$$

Вычисление такого следа не представляет никакой трудности, однако уже для следа 4  $\gamma$ -матриц:

$$\text{tr}(\gamma^\nu \gamma^\mu \gamma^\rho \gamma^\sigma) = 4(\eta^{\nu\mu} \eta^{\rho\sigma} - \eta^{\nu\rho} \eta^{\mu\sigma} + \eta^{\nu\sigma} \eta^{\mu\rho}) \quad (1.4)$$

ответ содержит 3 члена. Для следа 6  $\gamma$ -матриц - 15 членов, для 12 — 10395 членов.

Вычисление следов  $\gamma$ -матриц часто бывает необходимо для теоретико-полевых расчетов, но аналитически такие свертки высчитывать проблематично.

## История развития методов компьютерной алгебры

Впервые методы компьютерной алгебры для вычислений в КТП применил сам Фейнман. В 1962 году он выступал на конференции в Польше, где делал доклад,

посвященный попыткам построить квантовую теорию гравитации (которые не увенчались успехом).

Но, в частности, он упомянул, что алгебра для этой работы была подсчитана с помощью компьютера. Возможно, это было первое рабочее сложное вычисление, которое было полностью проведено на компьютере.

Буквально на следующий год в разных лабораториях появляются 3 системы компьютерной алгебры.

- 1) Самая известная — *schoonschip* (с голландского языка «почистить корабль»), разработанная Мартином Велтманом в 1963 году. В то время была популярна  $V - A$  («векторная минус аксиальная») теория слабого взаимодействия — предтеча современной электрослабой теории. В этой теории, в частности, впервые было показано, что в слабом взаимодействии нарушается пространственная четность. Для развития этой теории требовались громоздкие вычисления. Вдохновленный идеями Фейнмана, Велтман написал программу на языке Ассемблера для вычислений только в этой теории, но в последствии его программа была улучшена и стала полноценной компьютерной алгеброй. *Schoonschip* была разработана на компьютере IBM 7094. *Schoonschip* появился в СССР вместе с машиной CDC6600, доставленной в Дубну из за рубежа.
- 2) Система *Reduce* (1963), разработанная Энтони Херном, предназначенная для различных вычислений в КЭД. Написана на языке *LISP*, разработана на компьютере IBM 7090.
- 3) *Ashmedai* (1963), разработанная Майклом Левином для IBM 7090. Майкл Левин занимался вопросом, как нейтрино взаимодействуют в Солнце, для чего ему требовались сложные вычисления. Программа была написана на *FAP* (*fortran-assembly*).

Эти программы активно применялись для расчетов в задачах:

- Процесс  $e^+ + e^- \rightarrow W^- + W^+ \rightarrow e^- + \bar{\nu}_e + \mu^+ + \nu_\mu$ , требовавший описания и расчета на языке диаграмм Фейнмана.
- Вычисление аномального магнитного момента, требовавшее расчета в многопетлевом приближении. В рамках этой задачи стоит упомянуть работы Бродского - Киношиты 1969 и 1972 годов — в рамках которых было подсчитано 7 двухпетлевых диаграмм (см. рис. 1.2) на компьютере.
- Расчете Лэмбовского сдвига.

Через некоторое время стало понятно, что обходиться компьютерными вычислениями только для приведения подобных недостаточно — было необходимо брать интегралы Фейнмана, для чего были созданы свои программы. *SIN* была написана в 1966, и с ее помощью можно было брать интегралы.

Встал также вопрос о генерации диаграмм в заданном порядке. Пионерской работой в этой области является работа 1970 года Херна - Кэмпбелла, в которой они на компьютере генерировали диаграммы с помощью теоремы Вика. Главным недостатком их программы была невозможность идентификации одинаковых диаграмм.

В 80-е годы активно развивалась тема, связанная с вычислением интегралов:

- 1982 год — Четыркин - Ткачев. Они разработали метод, который позволял сводить сложные Фейнмановские интегралы к более простым Мастер-интегралам.
- 1984 год — была создана программа *MINCER* и с ее помощью была посчитана 5-петлевая  $\beta$ -функция.

Эти события на западе называли «Русской многопетлевой революцией».

**Замечание.** В дальнейшем в рамках курса мы будем работать в размерной технике регуляризации.

В 90-е — 00-ые появляются новые программы, позволяющие генерировать диаграммы:

- 1990 *Feynarts* — Кюбальбек, Бём, Дэйнер.
- 1992 *Qgraf* - Португальский ученый Пауло Ногейра. Эта программа позволяет генерировать диаграммы с использованием теории Графов.

Эти исследования привели к созданию нескольких научных журналов — « *Journal of Comp Phisics* » 1966, « *Computes Physics Computations* » 1969.

За последнее десятилетие появилось множество программ, методов для вычисления интегралов Фейнмана (разными методами), для вычисления сверток тензоров, генераторов событий.

## Философское отступление. Вопрос о «дегуманизации науки»

Некоторое время назад в ученой среде были активные рассуждения на тему «дегуманизации науки». Гуманно ли лишать людей возможности что-то посчитать руками?

В основе таких рассуждений лежит непонимание того, как работает человеческое сознание. Необходимо разделять работу, которая связана с «ремесленной» технической частью науки, и фундаментальные физические открытия (как, например, открытие Гейзенбергом квантовой механики).

В перспективе, техническую часть развития фундаментальной идеи можно отдать компьютеру, но «сказать принципиально новое слово в науке», выдать что-то абсолютно новое — компьютер не в состоянии.

## План Курса

- 1) Генерация диаграмм. Рассмотрим рекурсивный подход и подход, основанный на теории графов.
- 2) Генерация правил Фейнмана.
- 3) Вычисление интегралов
  - a) Сведение к мастер-интегралам.
  - b) Вычисление мастер-интеграла.
- 4) Работа в неабелевых теориях, алгебра генераторов. Например, в теориях  $SU(N)$  возникают генераторы алгебры в различных представлениях. В вычислениях часто бывает нужно вычислять следы генераторов в различных представлениях — используется пакет *color*.
- 5) Компьютерная алгебра в суперсимметричных теориях

## Лекция 2. Методы генерации диаграмм. Основы теории графов

Существуют 2 основные программы для генерации графов (диаграмм Фейнмана):

- 1) *Qgraf*, созданный в 1992, и написанный на языке *Fortran*
- 2) *Feynarts*, созданный в 1993, написанный с использованием системы компьютерной алгебры *Matematica*.

Все программы общего вида, производящие все этапы вычислений диаграмм (например, *FeynCalc*), до сих пор используют эти пакеты («нового слова» в вопросах генерации диаграмм с того времени сказано не было).

### Диаграммы Фейнмана. Производящий функционал

Рассмотрение генерации диаграмм начнем с понятия *производящего функционала* (можно начать и с теоремы Вика, это только вопрос удобства). Рассмотрим в качестве примера скалярную теорию с взаимодействием  $\lambda\varphi^4$ :

$$\mathcal{L} = \frac{1}{2} (\partial_\mu \varphi)^2 - \frac{1}{2} m^2 \varphi^2 - \frac{\lambda}{4!} \varphi^4 \quad (2.1)$$

Тогда производящий функционал

$$Z[j] = \exp \left\{ i \int d^4x \mathcal{L}_{\text{int}} \left( \frac{\delta}{i\delta J} \right) \right\} \exp \left\{ \frac{i}{2} \int d^4x \int d^4y j(x) D(x-y) j(y) \right\}, \quad (2.2)$$

где для нашей теории

$$\mathcal{L}_{\text{int}} = -\frac{\lambda}{4!} \varphi^4. \quad (2.3)$$

Разложим экспоненту со взаимодействием в ряд:

$$\begin{aligned} & \exp \left( i \int d^4x \left( -\lambda/4! \left( \frac{\delta}{i\delta J} \right)^4 \right) \right) \approx \\ & \approx 1 + i \int d^4x \left( -\frac{\lambda}{4!} \left( \frac{\delta}{i\delta J} \right)^4 \right) + \frac{1}{2} i^2 \int d^4x \int d^4y \frac{\lambda^2}{(4!)^2} \left( \frac{\delta}{i\delta x} \right)^4 \left( \frac{\delta}{i\delta y} \right)^4 \dots \end{aligned}$$

Сам по себе производящий функционал физического смысла не имеет, но из него можно получить функции Грина:

$$D(x-y) = \left. \frac{\delta}{i\delta j_x} \frac{\delta}{i\delta j_y} Z[j] \right|_{j=0} \quad (2.4)$$

Все члены этого ряда можно интерпретировать как диаграммы Фейнмана. Так, в первом порядке (однопетлевое приближение), в результате спаривания производных, получится диаграмма:

$$-i \frac{\lambda}{4!} \int d^4x \frac{\delta}{\delta j(x)} \frac{\delta}{\delta j(x)} \frac{\delta}{\delta j(x)} \frac{\delta}{\delta j(x)} Z_0 = 3\lambda \cdot \text{O}, \quad (2.5)$$

т.к. спаривание (горизонтальная линия) — пропагатор из точки, в которой берется первая производная в точку, в которой берется вторая. Фактор 3 можно получить из теоремы Вика, либо из соображений симметрии.

Рассмотрим второй порядок  $\lambda^2$ :

$$\int d^4x \int d^4y \frac{\delta}{\delta j_x} \frac{\delta}{\delta j_x} \frac{\delta}{\delta j_x} \frac{\delta}{\delta j_x} \frac{\delta}{\delta j_y} \frac{\delta}{\delta j_y} \frac{\delta}{\delta j_y} \frac{\delta}{\delta j_y} Z_0 \quad (2.6)$$

Для этого члена получается  $7 \cdot 5 \cdot 3 = 105$  вариантов спаривания. Однако реальных диаграмм будет всего 3 штуки (см. рис 2.1).

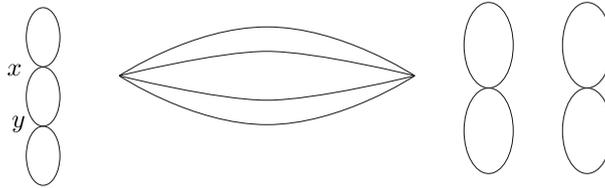


Рис. 2.1. Однопетлевые диаграммы в теории  $\lambda\phi^4$

В более сложном случае еще надо извлекать одночастично неприводимые диаграммы (которые в нашем простейшем случае отсутствуют). Но такой алгоритм — рассмотрение всевозможных спариваний, тоже иногда бывает полезен (например, в суперсимметричном случае), когда нужно определять симметричные факторы. Однако важно понимать, что такой топорный способ отнимает у компьютера очень много ресурсов.

### Вычисление симметричных коэффициентов диаграмм

Вычислим количество всевозможных спариваний для первой диаграммы на рисунке (2.1).

Так, первый  $\frac{\delta}{\delta j_x}$  может спариться с другим  $\frac{\delta}{\delta j_x}$  тремя способами, далее, второй  $\frac{\delta}{\delta j_x}$  может спариться с  $\frac{\delta}{\delta j_y}$  четырьмя способами, третий  $\frac{\delta}{\delta j_x}$  может спариться уже только с тремя  $\frac{\delta}{\delta j_y}$ , и две оставшиеся  $\frac{\delta}{\delta j_y}$  спариваются друг с другом. Таким образом, для этого случая  $3 \cdot 4 \cdot 3 = 36$  вариантов. (случай *a* рисунка 2.2)

Аналогично рассматриваем случай, когда  $\frac{\delta}{\delta j_x}$  сразу спаривается с  $\frac{\delta}{\delta j_y}$ , а второй  $\frac{\delta}{\delta j_x}$  спаривается с другим  $\frac{\delta}{\delta j_x}$  —  $4 \cdot 2 \cdot 3 = 24$  вариантов (случай *b* рисунка 2.2)

$$\begin{aligned}
 \text{a} \quad & \int d^4x \int d^4y \frac{\delta}{\delta j_x} \frac{\delta}{\delta j_x} \frac{\delta}{\delta j_x} \frac{\delta}{\delta j_x} \frac{\delta}{\delta j_y} \frac{\delta}{\delta j_y} \frac{\delta}{\delta j_y} \frac{\delta}{\delta j_y} Z_0 \\
 \text{b} \quad & \int d^4x \int d^4y \frac{\delta}{\delta j_x} \frac{\delta}{\delta j_x} \frac{\delta}{\delta j_x} \frac{\delta}{\delta j_x} \frac{\delta}{\delta j_y} \frac{\delta}{\delta j_y} \frac{\delta}{\delta j_y} \frac{\delta}{\delta j_y} Z_0 \\
 \text{c} \quad & \int d^4x \int d^4y \frac{\delta}{\delta j_x} \frac{\delta}{\delta j_x} \frac{\delta}{\delta j_x} \frac{\delta}{\delta j_x} \frac{\delta}{\delta j_y} \frac{\delta}{\delta j_y} \frac{\delta}{\delta j_y} \frac{\delta}{\delta j_y} Z_0
 \end{aligned}$$

Рис. 2.2. Всевозможные варианты спаривания в двухпетлевом приближении

Наконец,  $\frac{\delta}{\delta j_x}$  сразу спаривается с  $\frac{\delta}{\delta j_y}$ , и второй  $\frac{\delta}{\delta j_x}$  спаривается с  $\frac{\delta}{\delta j_y}$  (случай с рисунка 2.2) —  $4 \cdot 3 = 12$  вариантов.

Итого -  $36 + 24 + 12 = 72$ .

Количество вариантов спаривания для второй диаграммы —  $4! = 24$ .

Для третьей диаграммы, зная симметричный фактор каждой «восьмерки», получим  $3 \times 3 = 9$ .

В сумме, действительно, 105 вариантов спаривания. В этом заключается наивный подход к получению симметричных факторов.

## Существующие реализации генерации диаграмм. Программа *FeynArts*

Литература по теме:

- Публикация Т.Nahn (2001) в журнале CFC.
- Hagen Eck thesis — Development of a generic Feynman diagram generator (1995) - Feynarts.

Основная идея заключается в рекуррентной генерации. Рассмотрим графы с внешними линиями.

- 1) Берется начальная выборка с минимальным количеством внешних линий.
- 2) Выбирается шаг рекурсии — у диаграммы есть  $e - 1$  внешняя нога
- 3) Рассматриваются всевозможные варианты подстановки еще одной ноги, чтобы количество ног стало  $e - 1 \rightarrow e$ .

Изобразим алгоритм в виде блок-схемы (см. рис. 2.3).

**Замечание:** Под "топологией" будем понимать ненаполненную смыслом графическую «заготовку».

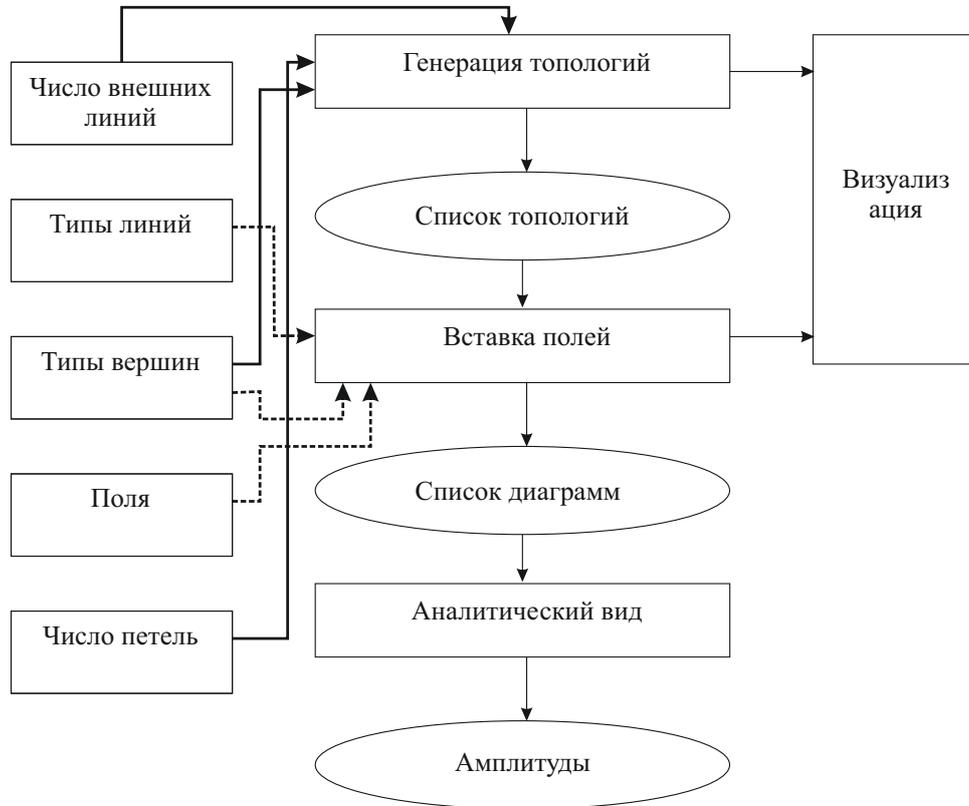


Рис. 2.3. Алгоритм работы программы *Feynarts*

### Пример генерации топологии в программе *FeynArts*

Рассмотрим самый простой вакуумный граф — окружность (См. рис 2.4). Будем рассматривать диаграммы в теории  $\lambda\phi^4$ .

- Чтобы получить из него более сложный граф, необходимо присоединить к нему пропагатор (всего 2 способа — горизонтальная стрелка «вправо»).
- Далее необходимо прибавить внешнюю линию. На каждом шаге рекурсии необходимо добавить линию либо к вершине, либо к пропагатору (стрелки «вниз»). После этого, проводится фильтрация, чтобы оставить только необходимые диаграммы.
- На следующих этапах вставляются типы вершин и полей.

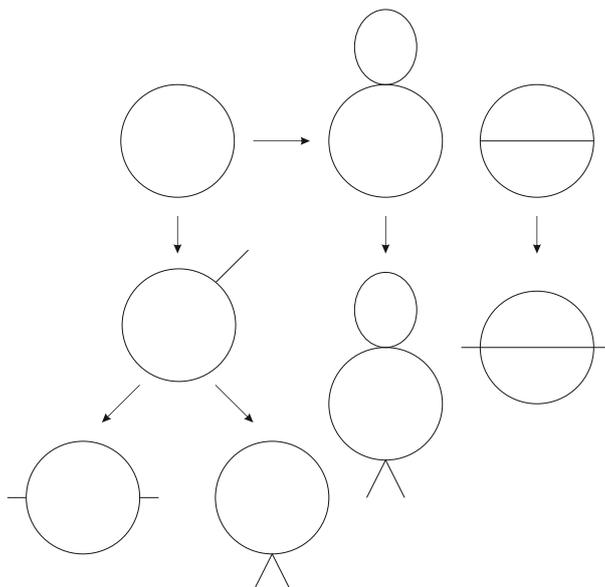


Рис. 2.4. Этапы генерации графов в программе *Feynarts* для теории  $\lambda\phi^4$

- После этого, на следующем шаге, можно произвести фильтрацию, в зависимости от рассматриваемой теории.

Программа ограничена 3-мя петлями, т.к. в более высоких порядках становится невозможно решить проблему изоморфизма графов — вычислить, какие графы равны между собой.

## Проблема изоморфизма графов

Как сравнить два графа между собой, если они выглядят различными? Например, рассмотрим случай одинаковых графов с различной нумерацией вершин — см. рис. 2.5. С точки зрения компьютера эти два графа различны — для сравнения

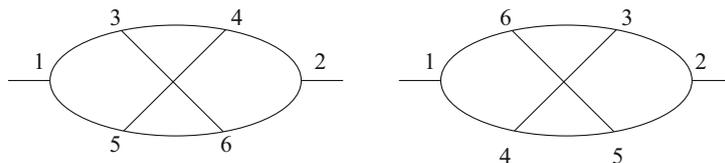


Рис. 2.5. Тожественные графы с различной нумерацией вершин

графов необходимо переставлять вершины друг с другом. Такая задача решается за неполиномиальное время (NP-время), т.е. число операций, за которое компьютер сравнивает два графа, растет экспоненциально в зависимости от числа вершин.

Проблемой тысячелетия считается задача доказательства того, что NP задача тождественна какой-то полиномиальной задаче.

## Теория графов. Генерация диаграмм в программе *Qgraf*

Классическая задача теории графов — **задача о 7 мостах Кёнигсберга**. Эту задачу разработал и решил Леонард Эйлер. Мосты между островами города можно представить в виде графа — см. рис 2.6. Возникает вопрос — можно ли пройти все 7

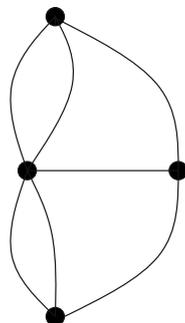


Рис. 2.6. К задаче о 7 мостах Кёнигсберга

мостов ровно по одному разу и вернуться в первоначальную точку. Эйлер доказал, что нет, введя т.н. *Эйлеровы циклы*.

Далее, теория развивалась, отделившись, наконец, в самостоятельную науку. Основные понятия в теории графов это:

- 1) Вершины  $v_i \in V(g)$
- 2) Ребра (линия, связывающая две вершины)  $e \in E(g)$ , например  $e = \{v_1, v_2\}$ .
- 3) Петля  $e = \{v, v\}$
- 4) Цикл - замкнутая линия.
- 5) Число циклов — цикломатическое число или контурный ранг. Удаляя по одной линии, можно привести граф к виду «дерева» — число удаленных линий равно числу петель.
- 6) Степень (валентность) вершины — количество линий, исходящих из вершины.

Графы делятся на несколько типов:

- 1) Простые графы
- 2) Графы с петлями — псевдографы.
- 3) Мультиграф — имеет набор одинаковых ребер  $\{e, e, e, \dots\}$ .
- 4) Псевдомультиграф — мультиграф с петлями.

Пример — см. рис. 2.7:

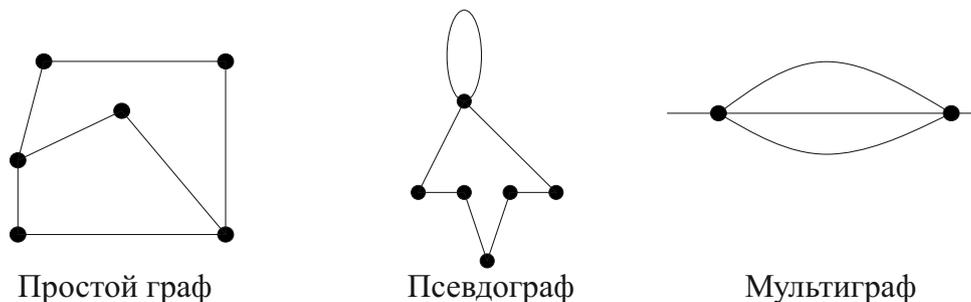


Рис. 2.7. Виды графов

### Способ задания графа при помощи матрицы смежности

Существует способ задания графа при помощи матрицы смежности  $v_{ij}$ , каждая компонента которой является количеством линии, соединяющих  $i$  и  $j$  вершину. Очевидно, матрица смежности всегда симметричная. Если  $i$ -тый диагональный элемент не равен нулю, то в этой вершине есть петля. Построим матрицу смежности для

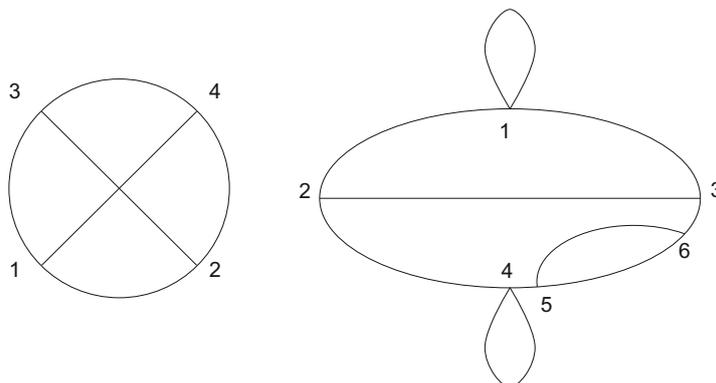


Рис. 2.8. К вопросу о матрице смежности

первого графа на рис. 2.8:

$$v_{ij} = \begin{pmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 \end{pmatrix} \quad (2.7)$$

Этот граф хорош тем, что как бы мы не переставляли вершины, матрица смежности будет та же самая.

Рассмотрим более сложный пример - второй граф на рис. 2.8:

$$v_{ij} = \begin{pmatrix} 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 2 \\ 0 & 0 & 1 & 0 & 2 & 0 \end{pmatrix} \quad (2.8)$$

- Переставляя столбцы и строки матриц смежности, можем сравнивать различные графы и таким образом проверить изоморфизм графов — *необходимо сравнить  $v!$  вариантов.*
- Можно пойти другим путем. Будем использовать понятие валентности вершин. Все графы можно отклассифицировать по последовательности степеней вершин. Рассмотрим валентности вершин второго графа рис 2.8:

$$\begin{array}{ccc} 1 : 4 & 2 : 3 & 3 : 3 \\ 4 : 4 & 5 : 3 & 6 : 3 \end{array}$$

Расположим их в порядке убывания  $(4, 4, 3, 3, 3, 3)$  — можно сравнивать графы по этой характеристике — у одинаковых графов эти последовательности должны быть одинаковыми. Эту простую проверку можно провести на первом этапе, чтобы сузить задачу.

- Способ с использованием т.н. *Нормального вида графа* — если взять все строчки матрицы смежности простого графа, и записать их последовательно, обрезав по диагонали, можно получить число в двоичной системе. Если переставлять строки друг относительно друга, можно получить максимальное число — это и есть нормальный вид графа.

**Пример получения нормального вида графа.** Рассмотрим первую диаграмму на рис. 2.8. Ее матрица смежности:

$$v_{ij} = \begin{pmatrix} 1 & 1 & 1 & 0 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 \\ 0 & 1 & 1 & 1 \end{pmatrix} \quad (2.9)$$

Выпишем строки последовательно, обрезая по диагонали:

$$\boxed{1110011011}$$

Можно проверить, что это и есть нормальный вид графа.

В программе *Qgraf* ее создатель Ногейра обошел проблему изоморфизма графов, генерируя всевозможные графы, но оставляя только те, которые сразу генерируются в нормальном виде. Таким образом, рассматриваются только заведомо уникальные диаграммы.

Литература по теме:

- 1) Noguera, 1992, *Qgraf*
- 2) Cameron, Coldbown, Rend, Wormald — «Cataloging the graphs on 10 vertices».

## Лекция 3. Демонстрация работы в программе *Qgraf*

В прошлый раз рассмотрели алгоритмы генерации диаграмм Фейнмана. На этом занятии мы рассмотрим работу в программе *Qgraf*.

### Установка программы *Qgraf*

- 1) Перейти по ссылке QGRAF's website Программе уже 32 года, но надо сказать, что она до сих пор обновляется, а сайт поддерживается.

Большинство современных пакетов, проделывающих все шаги — от генерации диаграмм до результатов петлевого интегрирования, в основном используют эту программу.

- 2) Во вкладке `downloads` скачиваем версию 3.6.10.

При скачивании может возникнуть проблема авторизации — после нажатия на ссылку может появиться окно, в котором необходимо авторизоваться. В этом случае можно вставить `anonymus` в качестве имени пользователя и пароля (опционально).

### Процесс работы в программе *Qgraf*

После распаковки архива и открытия соответствующей папки, можно посмотреть на файлы, которые в ней оказались. Эта «картина» может показаться необычной, для рядового пользователя.

Основным файлом программы является файл `qgraf-3.6.10.f08`. Конечно, полезно ознакомиться с документацией в файле `qgraf-3.6.10.pdf`.

**ВАЖНО:** Файл формата `fo8` является исполняемым файлом для языка *Fortran*. Для правильной работы программы необходимо скачать компилятор этого языка по ссылке <http://www.equation.com/servlet/equation.cmd?fa=fortran>. Для 64-битных систем рекомендуется установить версию `gcc-14.1.0-64.exe`.

Важно понимать, что это только компилятор — вместе с ним не идем никакой среды программирования на этом языке. Тем не менее для наших приложений (утилитарного использования программы) нам будет достаточно только компилятора.

- 1) Для компиляции наших файлов необходимо запустить командную строку (`Win + S`, после чего набрать команду `cmd`).
- 2) В командной строке указать путь к директории с программой (обычно, `cd downloads`).

3) В командной строке переходим в папку с программой — команда `cd qgraf-3.6.10`

4) Вводим команду `gcc qgraf-3.6.10.f08 -lgfortran -lquadmath`.

*Замечание:* Если после последней команды система написала сообщение «"gcc" не является внутренней или внешней командой, исполняемой программой или пакетным файлом», значит у вас не прописан параметр `Path` для компилятора `gcc`.

5) Возвращаемся в папку. Видим, что в ней появилось несколько новых файлов. Запускаем файл `a.exe`.

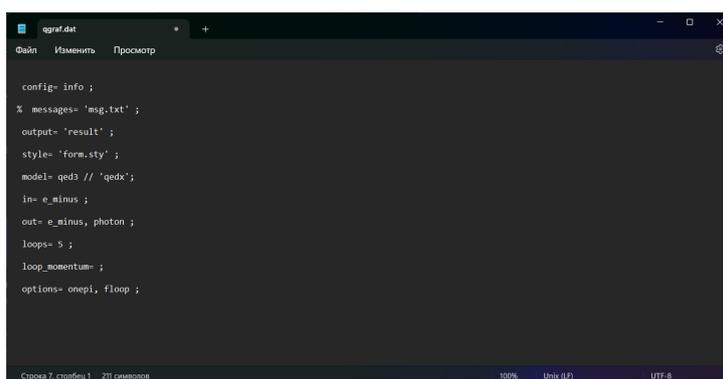
6) Открываем файл `qgraf.dat` (в блокноте).

В этом файле находятся все основные характеристики тех диаграмм, которые мы хотели бы получить.

7) Возле параметра `output` пишем название файла, в котором будет храниться результат генерации — например, можно вставить строчку `output= 'result' ;`

Перечислим другие параметры, которые находятся в файле `qgraf.dat`.

- `style` — стиль;
- `model` — выбор модели:
- `in` и `out` — входящие и выходящие поля;
- `loops` — число петель;
- `loop_momentum` — петлевой импульс;
- `options` — опции.



```
config= info ;
% messages= 'msg.txt' ;
output= 'result' ;
style= 'form.sty' ;
model= qed3 // 'qedx' ;
in= e_minus ;
out= e_minus, photon ;
loops= 5 ;
loop_momentum= ;
options= onepi, floop ;
```

Рис. 3.1. Внешний вид файла `qgraf.dat`

*Модели* находятся в файлах `qedx`, `qed`, `qcd` и `phi3`, в которых перечислены пропагаторы и вершины. В качестве примера, для теории  $\phi^3$  из файла `phi3`

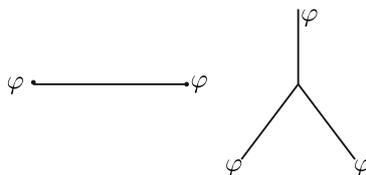


Рис. 3.2. Пропагаторы и вершины в теории  $\phi^3$

вершины и пропагаторы представлены на 3.2 Аналогично, для КЭД — в файле `qed`, открыв его в блокноте увидим:

```
% QED -- of electrons, positrons, and photons
[photon, photon, + ]
[e_minus, e_plus, - ]

[e_plus, e_minus, photon ]
```

Знаки  $+$  и  $-$  после пропагаторов очень важны — они показывают, фермионный ли это пропагатор или бозонный. От этих знаков зависит финальный результат (выполняется правильная коммутация полей для различных частиц). Кроме того, это позволяет учесть теорему Фарри, по которой если в петле на диаграмме нечетное количество фермионных вершин, то такая диаграмма равна нулю (см. рис. 3.3).

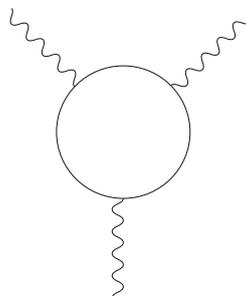


Рис. 3.3. Нулевая по теореме Фарри диаграмма

В модели `qedx` возникают целые наборы т.н. подмоделей (`submodel`). Они позволяют учесть 3 поколения фермионов в каждой из подмоделей. В результате выполнения программы с такой моделью будут генерироваться диаграммы со всевозможными фермионными петлями (даже с разными фермионами).

Программа выводит результат в двух **стилях** — стиль `form` (простой стиль) и стиль `array`, который показывает петлевые импульсы в каждом пропагаторе.

**Входящие поля** должны соответствовать тем полям, которые есть в модели.

Отдельно остановимся на **опциях**:

- `onery` — только одночастично неприводимые диаграммы.
- `onshell` — нет вставок на внешних линиях
- `nosigma` - вообще нет никаких собственноэнергетичных вставок (см. рис. 3.4).

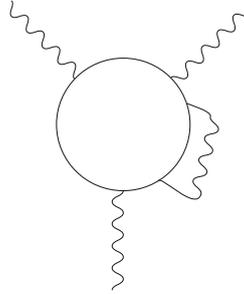


Рис. 3.4. Диаграмма с собственноэнергетической вставкой

- `simple` — один пропагатор соединяет 2 разных точки, и нет пропагаторов, начинающихся и заканчивающихся в одной точке (исключает мультиграфы и псевдографы).
  - `floor` — исключать случаи, зануляемые по теореме Фарри.
  - `topol` — рассмотрение только *топологий* (в смысле, в каком мы их понимали в предыдущей лекции).
- 8) Если хочется рассмотреть модель, которая отлична от предложенных выше, можно создать свой файл. Например, чтобы создать теорию  $\phi^4$ , необходимо создать файл 'phi4' и вписать соответствующие пропагаторы и вершины для нашей теории:

```
[ phi, phi, + ]
```

```
[ phi, phi, phi, phi ]
```

и после этого, вписать название модели в файл `qgraf.dat`.

- 9) Запускаем программу `a.exe`.
- 10) В новом файле `result` представлен результат генерации.

```

#
# file generated by qgraf-3.6.10
#
# config= info ;
# output= 'result' ;
# style= 'form.sty' ;
# model= 'phi4';
# in= phi ;
# out= phi ;
# loops= 1 ;
# loop_momentum= k;
# options= onepi ;
#
*--#[ d1:
*
    1/2
    *vx(phi(-1),phi(-2),phi(1),phi(1))
*
*--#] d1:
#
# end
#

```

Попробуем проинтерпретировать этот результат. В самом начале файла, программа «напоминает», какие изначально были опции. Далее идут диаграммы (в данном примере диаграмма одна d1).

По сути, программа перечисляет, какие вершины у нас есть. В начале идет симметричный коэффициент  $\frac{1}{2}$ . Отрицательными числами описываются внешние линии. Положительными числами нумеруются точки и места соединения двух линий. Результат генерации диаграммы представлен на рисунке 3.5. Симметричный коэффициент получился таким, т.к. теория уже учитывает  $4!$  из члена, который ответственен за взаимодействие —  $-\frac{\lambda}{4!}\phi^4$ .

Тогда, т.к. поля  $\phi_x\phi_x\phi_x\phi_x$  спариваются 12- способами (4 возможности для первой внешней линии, и 3 — для другой), как раз получим:

$$\frac{12}{4!} = \frac{1}{2}$$

11) Для того, чтобы получить новый результат, необходимо удалить предыдущий

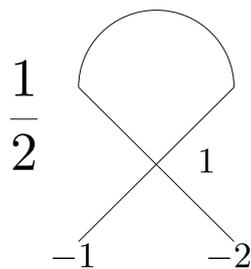


Рис. 3.5. Сгенерированная диаграмма в теории  $\phi^4$

(программа не перезаписывает файлы).

## Различные примеры генерации графов в программе *Qgraf*

- 1) Для той же теории, но для 2-ух петель (можно изменить параметр `loops`) в конфигурационном файле, получим

```
--#[ d1:
*
  1/4
  *vx(phi(-1),phi(-2),phi(1),phi(2))
  *vx(phi(1),phi(2),phi(3),phi(3))
*
*--#[ d1:
*--#[ d2:
*
  1/6
  *vx(phi(-1),phi(1),phi(2),phi(3))
  *vx(phi(-2),phi(1),phi(2),phi(3))
*
*--#[ d2:
#
# end
#
```

Видим, что получили 2 диаграммы — см. рис. 3.6. Из рассмотрения, аналогичного уже произведенному, можем убедиться в правильности симметричных коэффициентов.

- 2) Рассмотрим ситуацию 4-точечной функции и припишем импульсы. Для этого в конфигурационном файле меняем строчки, ответственные за входящие и

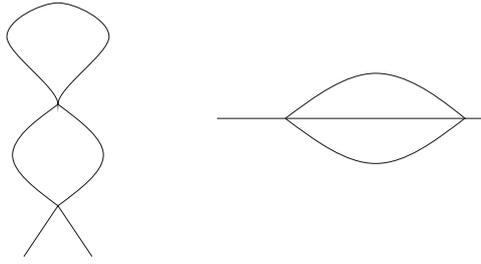


Рис. 3.6. Сгенерированные диаграммы в теории  $\phi^4$  в 2-ух петлях

исходящие линии на

```
in= phi [p1], phi [p2] ;
```

```
out= phi [q1], phi [q2] ;
```

В результате получится 12 диаграмм. Однако некоторые из этих диаграмм будут повторяться.

- 3) Если применить опцию `topol`, можно проверить, что на самом деле всего существует только 3 топологии (см. рис. 3.7).

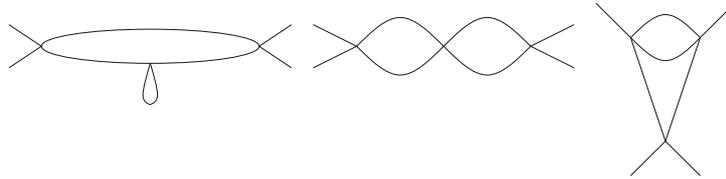


Рис. 3.7. Сгенерированные диаграммы в теории  $\phi^4$  в 2-ух петлях для 4 внешних линий

- 4) Рассмотрим также одночастично приводимые диаграммы. Для этого в конфигурационном файле напишем:

```
options= onep, topol ;
```

Получаем 5 различных диаграмм. Появились также такие варианты, как на рисунке 3.8.

- 5) поменяем стиль на `array`. Вставим в конфигурационный файл команду:

```
style= 'array.sty' ;
```

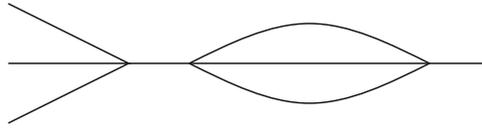


Рис. 3.8. Одночастично неприводимая диаграмма в теории  $\varphi^4$

Для простоты изменим количество внешних линий — 2 линии:

```
in= phi;
out= phi;
```

и будем рассматривать все диаграммы одночастично неприводимые:

```
options= onepi;
```

Получаем 2 диаграммы в новом представлении:

```
a(1):= (+1/4)*
pol(phi(-1,p1))*
pol(phi(-2,q1))*
prop(phi(1,k1))*
prop(phi(3,-k1))*
prop(phi(5,k2))*
vrtx(phi(-1,p1),phi(-2,-q1),phi(1,k1),phi(3,-k1))*
vrtx(phi(2,-k1),phi(4,k1),phi(5,k2),phi(6,-k2));

a(2):= (+1/6)*
pol(phi(-1,p1))*
pol(phi(-2,q1))*
prop(phi(1,k1+k2-p1))*
prop(phi(3,-k1))*
prop(phi(5,-k2))*
vrtx(phi(-1,p1),phi(1,k1+k2-p1),phi(3,-k1),phi(5,-k2))*
vrtx(phi(-2,-q1),phi(2,-k1-k2+p1),phi(4,k1),phi(6,k2));

% end
```

- pol — внешние линии;
- prop — пропагаторы;

- `VRTX` — вершины.

Графическое представление диаграмм показано на рисунке 3.9. Программа,

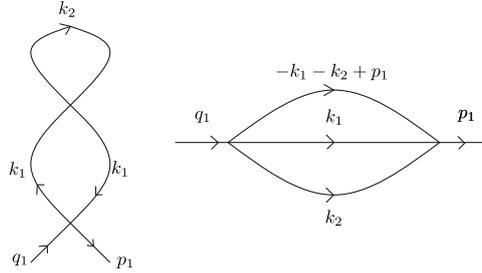


Рис. 3.9. Диаграммы сгенерированные в стиле `array`

правда, не учитывает ЗСЭ, необходимо задавать его самостоятельно ( $q_1 = p_1$ ).

- Рассмотрим ситуацию, связанную с КЭД. Построим диаграмму, дающую аномальный магнитный момент. В конфигурационном файле меняем модель, задаем внешние линии, количество петель, рассматриваем диаграммы в стиле `form`:

```
style= 'form.sty' ;
model= 'qed';
in= e_minus, e_plus;
out= photon;
loops= 1 ;
```

Можно убедиться, что генерируется нужная диаграмма (см. рис 1.1):

```
*--#[ d1:
*
  1
  *vx(e_plus(2),e_minus(-1),photon(1))
  *vx(e_plus(-3),e_minus(3),photon(1))
  *vx(e_plus(3),e_minus(2),photon(-2))
*
*--#[ d1:
#
# end
```

- 7) Можно рассмотреть и более сложную ситуацию для фермионов 3-х поколений — `model = qed3 // 'qedx'`, учитывая теорему Фарри — `options = onepi, floop` в 2-ух петлях.

Можно убедиться, что в результате получится 9 диаграмм — 7 для аномального магнитного момента из лекции 1 (см. рис. 1.2), и еще 2 для мюонной и таонной петель в центре.

## Лекция 4. Демонстрация работы в программе *FeynArts*

Пакет *FeynArts* является более дружелюбной к пользователю утилитой, чем программа *Qgraf*, которая была рассмотрена ранее. *FeynArts* является пакетом для *Mathematica*, а значит чтобы пользоваться этим пакетом необходимо установить сначала эту программу.

### Установка пакета *FeynArts*

- 1) Перейдем на сайт <https://feynarts.de/>. Здесь можно найти ссылки на оригинальные статьи, посвященные этой программе, и ссылку на скачивание этой программы.

Спускаясь ниже по странице, можем увидеть различные версии программы, — программа регулярно обновляется.

На момент записи лекции, самая новая версия программы `FeynArts-3.12.tar.gz`.

*Замечание:* Особенностью этого пакета является критическая зависимость этого пакета от конкретной модели. Если в *Qgraf* можно самим задавать вершины и пропагаторы теории, учитывая даже грассманову четность, то в *FeynArts* модели «защиты» в программу.

Для подробного ознакомления с возможностями программы и теорией можно обратиться к диссертации Hagen Eck's Ph.D. thesis, ссылка на которую представлена на той же странице.

- 2) Скачанный архив надо разархивировать в специальную папку, в которой хранятся пакеты для *Mathematica*. В стандартном случае путь такой `user/AppData/Roaming/Mathematica/Applications`.
- 3) Далее, необходимо открыть *Mathematica*, создать новый документ и ввести команду `<< Feynarts.m`. После этого пакет можно использовать.

### Работа с пакетом *FeynArts*

Суть работы программы представлена на рисунке 2.3. Сначала генерируются топологии, потом в эти топологии вставляются поля, потом генерируются диаграммы. В конце, программа может построенные диаграммы даже нарисовать.

Возможность этого пакета, которая выгодно отличает его от программы *Qgraf* — построение амплитуд процессов.

1) Первым этапом надо построить топологии.

- При генерации топологий не учитываются никакие правила (Фейнмана или другие). Синтаксис:

```
CreateTopologies[l, i -> o]
```

где первая цифра  $l$  — число петель,  $i$  — количество входящих полей,  $o$  — количество выходящих полей.

В качестве примера, рассмотрим случай диаграммы с одной входящей линией, одной выходящей линией и одной петлей — `CreateTopologies[1, 1 -> 1]`.

Вывод программы:

```
TopologyList[
Topology[2] [
Propagator[Incoming][Vertex[1][1], Vertex[4][3]],
Propagator[Outgoing][Vertex[1][2], Vertex[4][3]],
Propagator[Loop[1]][Vertex[4][3], Vertex[4][3]] ]
Topology[2] [
Propagator[Incoming][Vertex[1][1], Vertex[3][3]],
Propagator[Outgoing][Vertex[1][2], Vertex[3][3]],
Propagator[Internal][Vertex[3][3], Vertex[3][4]],
Propagator[Loop[1]][Vertex[3][4], Vertex[3][4]] ],
Topology[2] [
Propagator[Incoming][Vertex[1][1], Vertex[3][3]],
Propagator[Outgoing][Vertex[1][2], Vertex[3][4]],
Propagator[Loop[1]][Vertex[3][3], Vertex[3][4]],
Propagator[Loop[1]][Vertex[3][3], Vertex[3][4]] ] ]
```

Каждой вершине соответствует объект вида `Vertex[adj][n]`, где `[adj]` — число входящих в нее пропагаторов, а `[n]` — номер этой вершины. `Loop[n]` - петля с номером `[n]`. Первая диаграмма представлена на рисунке 4.1.

2) Диаграммы можно нарисовать и в самой программе — команда `Paint[%]`. Вывод программы показан на рисунке 4.2.

- В рамках пакета есть возможность учитывать также контрчлены, т.е. диаграммы, ответственные за перенормировки.

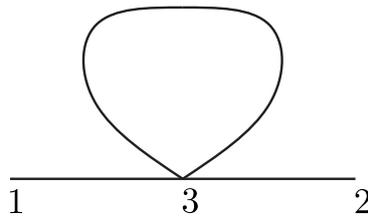


Рис. 4.1. Первая сгенерированная диаграмма

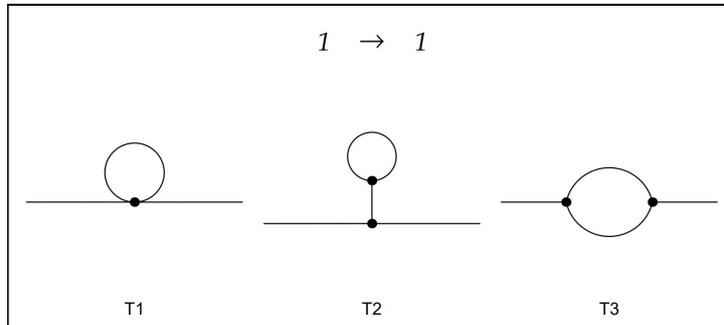


Рис. 4.2. Диаграммы, сгенерированные и нарисованные пакетом *FeynArts*

- Можно также исключать из рассмотрения различные диаграммы. Так, команда `Reducible` исключает из рассмотрения одночастично приводимые диаграммы.
- 3) Вторым этапом, необходимо вставить поля. Обозначения полей —  $F$ ,  $S$ ,  $V$ ,  $U$ ,  $T$ , соответственно фермионные, скалярные, векторные, духовые и тензорные поля. Поля вставляются командой `InsertFields[t, {i1, i2, ...} -> {o1, o2, ...}]`, где  $t$  — название переменной, в которой хранится топология, которую мы определили ранее,  $\{i1, i2, \dots\}$  - входящие поля,  $o1, o2, \dots$  — выходящие поля.
- 4) Генерация работает в рамках какой-то модели (по умолчанию — *Стандартная модель без хромодинамики*). Описание модели можно найти в конце мануала. Так, для диаграммы собственной энергии электрона, код программы будет (здесь и далее, в силу большого масштаба выходных данных, вывод программы не приводится. Рекомендуется самостоятельно запускать код из примеров):

```
t12 = CreateTopologies[1, 1 -> 1,
ExcludeTopologies -> Internal];
InsertFields[ t12, F[2, {1}] -> F[2, {1}]];
```

- 5) Аналогичным образом, можно рассмотреть процесс с одним входящим электроном, и выходящими электроном и фотоном:

```
t12 = CreateTopologies[1, 1 -> 2,  
ExcludeTopologies -> Internal];  
InsertFields[ t12, F[2, {1}] -> {F[2, {1}], V[1]}];
```

- 6) Модель генерации диаграмм задается на моменте вставки полей. Чтобы поменять модель, в которой работает программа, необходимо в команду `InsertFields` добавить параметр `Model`. Так, для предыдущего примера, если мы хотим работать в рамках Квантовой Электродинамики (QED) команда будет (надо также изменить названия полей на те, которые соответствуют модели):

```
InsertFields[ t12, F[1] -> {F[1], V[1]}, Model -> "QED"];
```

- 7) Можно также исключать какие-то типы вершин или полей. Например, чтобы для предыдущего случая оставить только диаграммы, содержащие фотонные линии (работаем в рамках CM), можно воспользоваться командой:

```
InsertFields[ t12, F[1] -> {F[1], V[1]},  
LastSelections -> {V[1]}];
```

- 8) *FeynArts* содежит в себе также пакет для  $\text{\LaTeX}$  для рисования диаграмм Фейнмана.
- 9) Отдельного внимания заслуживает также модель, включающая хромодинамику. Чтобы построить трехглюонную вершину, код может быть:

```
t12 = CreateTopologies[1, 1 -> 2,  
ExcludeTopologies -> Internal];  
T = InsertFields[ t12, V[5, {1}] -> {V[5, {2}], V[5, {3}]},  
Model -> "SMQCD" ];  
Paint[T]
```

- В результате анализа выходных данных становится понятно, что при расчете амплитуды процесса необходимо вычислять следы от сверток генераторов алгебры  $su(3)$  в различных представлениях.

- Кроме того, поскольку сами кварки являются фермионами, в ответе присутствуют  $\gamma$ -матрицы — например,  $ga(\text{Lor3})$ .  $\text{Lor3}$  - третий Лоренцевый индекс. Возникают свертки этих матриц.
- Сам пакет *FeynArts* вычислять такие следы и свертки **не умеет**. Необходимо пользоваться сторонними пакетами.

## Расчет амплитуд процессов в программе *FeynArts*

. Код для расчета амплитуд для предыдущего процесса:

```
t12 = CreateTopologies[1, 1 -> 2,  
ExcludeTopologies -> Internal];  
T = InsertFields[ t12, F[1] -> {F[1], V[1]},  
LastSelections -> {V[1]}];  
CreateFeynAmp[T]
```

Следующим этапом, можно «попросить программу» вывести амплитуду в более простом представлении, или в форме, в которой амплитуды задаются для других программ компьютерной алгебры.

## Ограничения пакета *FeynArts*

Если попытаться выполнить код

```
t12 = CreateTopologies[4, 1 -> 2,  
ExcludeTopologies -> Internal];
```

программа выдаст ошибку. Дело в том, что программа может генерировать топологии с количеством петель не более 3 включительно.

Стоит упомянуть относительно большое время генерации диаграмм программой *FeynArts*, особенно по сравнению с *Qgraf*.

## Лекция 5. Представления петлевых интегралов Фейнмана

Литература по теме:

- 1) *Evaluating Feynman Integrals — Smirnov, 2007*. Книга считается устаревшей, потому что за последнее десятилетие было сделано много новых открытий в этой теме.
- 2) *Feynman Integrals — Springer, 2023*. Книга отражает все современные методы вычисления интегралов.

### Общий подход к рассмотрению диаграмм

- 1) Получить диаграмму;
- 2) Получить амплитуды;
- 3) Получить петлевые интегралы Фейнмана.

Здесь и далее речь идет именно о *петлевых интегралах*, а не об интегралах по фазовому пространству.

Если в одной петле интеграл берется вручную, то для высших же петель вычисление интегралов вручную является затруднительным.

Интегралы будем рассматривать **регуляризованными**. *Золотым стандартом* считается **размерная регуляризация**, — мы также будем исповедовать этот подход в курсе.

### Простейший пример вычисления петлевых интегралов

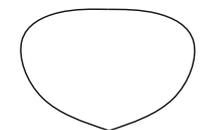

$$\sim \int \frac{d^4 k}{(2\pi)^4} \frac{1}{k^2 - m^2}$$

Рис. 5.1. Простейший пример петлевого интеграла

Самый простой пример — интеграл в теории  $\phi^4$  диаграммы (см. рис 5.1).

Производим поворот Вика и переходим от размерности 4 к размерности  $D$ :

$$\begin{aligned} \int \frac{d^4 k}{(2\pi)^4} \frac{1}{k^2 - m^2} &\rightarrow - \int i \frac{d^4 k_E}{(2\pi)^4} \frac{1}{k_E^2 + m^2} \\ &\rightarrow -i \int \frac{d^D k_E}{(2\pi)^D} \frac{1}{k_E^2 + m^2} \end{aligned} \quad (5.1)$$

Далее, считаем  $D = 4 - 2\varepsilon$ , где  $\varepsilon$  — малая величина.

В зависимости от числа измерений значение такого интеграла будет разным. На этом принципе и основан метод размерной регуляризации — переходя к дробной размерности и вычисляя интеграл в ней (в такой размерности интеграл сходится), можно затем перейти обратно к 4-ем измерениям, приняв  $\varepsilon \rightarrow 0$ .

Конечно, хотелось бы найти алгоритм, позволяющий сводить интегралы к более простым согласно этой технике. Такой алгоритм есть — *IBP-reduction*, техника мастер-интегралов, о нем пойдет речь в следующей лекции.

## Примеры интегралов Фейнмана

Рассмотрим интеграл  $T_V$ . Можно сказать, что это более общий случай интеграла (5.1)

$$T_V = e^{\varepsilon\gamma_E} (\mu^2)^{v-\frac{D}{2}} \int \frac{d^D k}{i\pi^{D/2}} \frac{1}{(-k^2 + m^2)^v}, \quad (5.2)$$

где

- $\gamma_E \approx 0.577$  - постоянная Эйлера, введённая для удобства;
- $\mu^2$  — размерный параметр, который всегда вводится в размерной технике, т.к. при переходе к  $D$  размерностям меняется размерность всего меры интеграла  $d^D k_E$ , и его нужно скомпенсировать.
- $\pi^{D/2}$  — введена для удобства
- $i$  — введена, чтобы удобно было перейти к Евклидову пространству.

Сначала производим поворот Вика — переходим к Евклидову пространству

$$T_V \rightarrow e^{\varepsilon\gamma_E} (\mu^2)^{v-\frac{D}{2}} \int \frac{d^D k}{\pi^{D/2}} \frac{1}{(k^2 + m^2)^v}. \quad (5.3)$$

Переходим в сферическую систему координат —  $d^D k = k^{D-1} dk d\Omega$ . Т.к. интеграл сферически-симметричный, можем взять интеграл по углам — в  $D$  измерениях справедлива формула:

$$\int d\Omega = \frac{2\pi^{D/2}}{\Gamma(\frac{D}{2})}. \quad (5.4)$$

Изменим меру интегрирования как

$$d^D k = \frac{1}{2} k^{D-2} dk^2 d\Omega = \frac{1}{2} (k^2)^{D/2-1} dk^2 d\Omega \quad (5.5)$$

Тогда интеграл

$$\begin{aligned} T_\nu &= e^{\varepsilon\gamma_E} (\mu^2)^{\nu-D/2} \frac{1}{\Gamma(\frac{D}{2})} \int_0^{+\infty} \frac{(k^2)^{D/2-1}}{(k^2+m^2)^\nu} dk^2 = \\ &= e^{\varepsilon\gamma_E} (\mu^2)^{\nu-D/2} \frac{1}{\Gamma(\frac{D}{2})} \underbrace{\frac{1}{m^{2\nu}} \cdot (m^2)^{D/2-1} \cdot m^2}_{=(m^2)^{D/2-\nu}} \int_0^{+\infty} \frac{t^{D/2-1}}{(1+t)^\nu} dt \end{aligned} \quad (5.6)$$

где  $t = \frac{k^2}{m^2}$ . Введем бета-функцию Эйлера  $B(z_1, z_2)$ :

$$B(z_1, z_2) = \int_0^\infty \frac{t^{z_1-1}}{(1+t)^{z_1+z_2}} dt \quad (5.7)$$

$$B(z_1, z_2) = \frac{\Gamma(z_1)\Gamma(z_2)}{\Gamma(z_1+z_2)} \quad (5.8)$$

Тогда, сравнивая коэффициенты, получим

$$T_\nu = e^{\varepsilon\gamma_E} \left(\frac{\mu^2}{m^2}\right)^{\nu-D/2} \frac{1}{\Gamma(\frac{D}{2})} \cdot \frac{\Gamma(\frac{D}{2})\Gamma(\nu-\frac{D}{2})}{\Gamma(\nu)} = \quad (5.9)$$

$$= e^{\varepsilon\gamma_E} \left(\frac{\mu^2}{m^2}\right)^{\nu-D/2} \cdot \frac{\Gamma(\nu-\frac{D}{2})}{\Gamma(\nu)} \quad (5.10)$$

Рассмотрим случай  $\nu = 1$  (по сути, это соответствует случаю, который мы рассмотрели ранее, см. рис. 5.1):

$$\begin{aligned} T_1 &= e^{\varepsilon\gamma_E} \left(\frac{\mu^2}{m^2}\right)^{1-D/2} \frac{\Gamma(1-\frac{D}{2})}{\Gamma(1)} = \\ &= \frac{m^2}{\mu^2} e^{\varepsilon\gamma_E} \Gamma(-1+\varepsilon) e^{-\varepsilon L} \end{aligned} \quad (5.11)$$

где  $L = \ln \frac{m^2}{\mu^2}$ . Далее, раскладывая в ряд, получим

$$T_1 = \frac{m^2}{\mu^2} \left[ -\frac{1}{\varepsilon} + (L-1) + o(\varepsilon) \right] \quad (5.12)$$

Конечно, для более сложных интегралов такой подход работать не будет. Чтобы описывать такие интегралы необходимо использовать другие представления.

## Представления интегралов

Рассмотрим интеграл вида:

$$I = e^{l\varepsilon\gamma_E} (\mu^2)^{\nu-ID/2} \int \prod_{r=1}^l \frac{d^D k_r}{i\pi^{D/2}} \prod_{j=1}^{n_{int}} \frac{1}{(-q_j^2 + m_j^2)^{\nu_j}}, \quad (5.13)$$

где  $l$  - число петель,

$$q_j = \sum_{r=1}^l \lambda_{jr} k_r + \sum_{r=1}^{n_{ext}} \sigma_{jr} p_r, \quad (5.14)$$

$k_r$  — петлевые импульсы,  $n_{ext}$  — число внешних линий,  $p_r$  — внешние импульсы,

$$\nu = \sum_j^{n_{int}} \nu_j. \quad (5.15)$$

Этот интеграл можно считать наиболее общим петлевым интегралом Фейнмана. Как мы можем с этим работать? Необходимо перейти в другое представление.

### Представление Швингера

Рассмотрим формулу:

$$\boxed{\frac{1}{A^\nu} = \frac{1}{\Gamma(\nu)} \int_0^\infty d\alpha \alpha^{\nu-1} e^{-\alpha A}} \quad (5.16)$$

Все подобного рода члены в произведении (5.13) можно представить с помощью этой формулы в виде интеграла от экспоненты, и тогда произведение таких множителей перейдет в сумму под экспонентой:

$$\frac{1}{(-q_j^2 + m_j^2)^{\nu_j}} = \frac{1}{\Gamma(\nu_j)} \int_{\alpha_j > 0} d\alpha_j \alpha_j^{\nu_j-1} \exp\{-\alpha_j(-q_j^2 + m_j^2)\} \quad (5.17)$$

Тогда

$$I = \frac{e^{l\varepsilon\gamma_E} (\mu^2)^{\nu-ID/2}}{\prod_{j=1}^{n_{int}} \Gamma(\nu_j)} \int_{\alpha_j > 0} d^{n_{max}} \alpha \left( \prod_{j=1}^{n_{max}} \alpha_j^{\nu_j-1} \right) \times \\ \times \int \prod_{r=1}^l \frac{d^D k_r}{i\pi^{D/2}} \exp\left\{-\sum_{j=1}^{n_{int}} \alpha_j(-q_j^2 + m_j^2)\right\} \quad (5.18)$$

где  $n_{max}$  — число импульсов, т.е.  $j = 1 \dots n_{max}$ . Таким образом, все знаменатели мы занесли под экспоненту.

Рассмотрим отдельно выражение, стоящее под экспонентой. Его можно представить как свертку петлевых импульсов  $k_i$ :

$$\sum_{j=1}^{n_{max}} \alpha_j (-q_j^2 + m_j^2) = \sum_j^l \sum_n^l k_j M_{jn} k_n + 2 \sum_j^l k_j v_j + C, \quad (5.19)$$

где  $M_{jn}$  — числовая матрица размерности  $n \times n$ . Все эти матрицы должны содержать в себе параметры  $\alpha$ . Введем т.н. *полиномы Симанзика*:

$$U = \det(M), \quad F = \det M \times (v^T M^{-1} v + C) / \mu^2. \quad (5.20)$$

Также можно доказать следующую формулу

$$\int_{-\infty}^{\infty} dy_1 \dots dy_n \exp \{-y^T A y + 2x^T y + j\} = \pi^{N/2} \frac{1}{(\det A)^{1/2}} \exp \{x^T A^{-1} x + j\}. \quad (5.21)$$

Формула вычисления интеграла Гаусса является одномерной аналогией формулы выше (5.21):

$$\int_{-\infty}^{\infty} e^{-ax^2 + bx + c} dx = e^{-(\sqrt{ax-b/\sqrt{a}})^2 + c + b^2/a} = \sqrt{\frac{\pi}{a}} e^{b \cdot \frac{1}{a} b + c} \quad (5.22)$$

Полиномы Симанзика являются полиномами  $l$ -той степени по  $\alpha$ , кроме того они оказываются однородными со степенью однородности  $l$  и  $l+1$  для  $U$  и  $F$ , то есть

$$U(td) = t^l U(d), \quad \text{где } t = \text{const} \quad (5.23)$$

$$F(td) = t^{l+1} F(d) \quad (5.24)$$

Теперь применяем эти формулы к преобразованию интеграла  $I$  (Виков поворот уже произвели):

$$I = \frac{e^{l\epsilon\gamma_E} (\mu^2)^{v-ID/2}}{\prod_{j=1}^{n_{max}} \Gamma(v_j)} \int_{\alpha_j > 0} d^{n_{max}} \alpha \left( \prod_{j=1}^{n_{max}} \alpha_j^{v_j-1} \right) \frac{\pi^{D/2}}{\pi^{D/2}} \frac{1}{(U(\alpha_j))^{D/2}} \exp \left\{ -\frac{F}{U} \right\} \quad (5.25)$$

### Представление Фейнмана

Будем использовать формулу:

$$\boxed{\int_{-\infty}^{\infty} dt \delta \left( t - \sum_{i=1}^{n_{int}} \alpha \right) = 1}, \quad \text{если } \alpha > 0 \quad (5.26)$$

На этой формуле основано следующее утверждение. Рассмотрим функцию  $f(\alpha_1, \dots, \alpha_n)$ , представим  $\alpha$  как  $\alpha = t \cdot a$ , тогда

$$\int d^n \alpha_n f(\alpha_1, \dots, \alpha_n) = \int_{a_j > 0} d^n a t^n f(ta_1 \dots ta_n) = \quad (5.27)$$

$$= \int d^n a \delta \left( 1 - \sum_{j=1}^n a_j \right) \int dt t^{n-1} f(ta_1 \dots ta_n) \quad (5.28)$$

Последнее утверждение нужно, чтобы перейти к представлению Фейнмана от представления Швингера:

$$\begin{aligned}
 I &= \frac{e^{l\varepsilon\gamma_E} (\mu^2)^{\nu-ID/2}}{\prod_{j=1}^{n_{max}} \Gamma(\nu_j)} \int d^{n_{max}} \alpha \left( \prod_{j=1}^{n_{max}} a_j^{\nu_j-1} \right) \delta \left( 1 - \sum_{j=1}^{n_{max}} a_j \right) \frac{1}{U(\alpha_j)^{D/2}} \times \\
 &\times \int_0^\infty dt t^{\nu-\frac{ID}{2}-1} \exp \left\{ -\frac{F}{U} t \right\} = \\
 &= \frac{e^{l\varepsilon\gamma_E} (\mu^2)^{\nu-ID/2}}{\prod_{j=1}^{n_{max}} \Gamma(\nu_j)} \int_{\alpha_j > 0} d^{n_{max}} \alpha \left( \prod_{j=1}^{n_{max}} a_j^{\nu_j-1} \right) \delta \left( 1 - \sum_{j=1}^{n_{max}} a_j \right) \frac{U^{\nu-(\frac{l+1}{2})D}}{F^{\nu-ID/2}} \times \\
 &\times \underbrace{\int_0^\infty dt t^{\nu-\frac{ID}{2}-1} \exp(-t)}_{\Gamma(\nu-\frac{ID}{2})}
 \end{aligned} \tag{5.29}$$

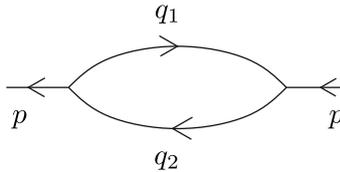
Таким образом, в представлении Фейнмана

$$I = \frac{e^{l\varepsilon\gamma_E} \Gamma(\nu - \frac{D}{2})}{\prod_{j=1}^{n_{max}} \Gamma(\nu_j)} \int_{\alpha_j > 0} d^{n_{max}} a \delta \left( 1 - \sum_{j=1}^{n_{max}} a_j \right) \prod_{j=1}^{n_{max}} a_j^{\nu_j-1} \frac{U^{\nu-\frac{l+1}{2}D}}{F^{\nu-\frac{ID}{2}}} \tag{5.30}$$

$\mu^2$  в соответствующей степени ушло, т.к. она есть в определении полинома  $F$ .

## Пример вычисления петлевого интеграла с использованием различных представлений

Диаграмма представлена на рис 5.2.



$$\sim I_{\nu_1 \nu_2} = e^{\varepsilon\gamma_E} (\mu^2)^{\nu-D/2} \int \frac{d^D k}{i\pi^{D/2}} \frac{1}{(-q_1^2 + m^2)^{\nu_1} (-q_2^2 + m^2)^{\nu_2}}$$

Рис. 5.2. Пример петлевого интеграла общего вида

$$\sim I_{\nu_1 \nu_2} = e^{\varepsilon\gamma_E} (\mu^2)^{\nu-D/2} \int \frac{d^D k}{i\pi^{D/2}} \frac{1}{(-q_1^2 + m^2)^{\nu_1} (-q_2^2 + m^2)^{\nu_2}} \tag{5.31}$$

Учтем закон сохранения  $p = q_2 - q_1$ . Введем

$$q_2 = p + k \tag{5.32}$$

$$q_1 = k \tag{5.33}$$

Тогда

$$\begin{aligned} \sum_{j=1}^{n_{\max}} \alpha_j (-q_j^2 + m_j^2) &= \alpha_1 (-k^2 + m_1^2) + \alpha_2 (-(p+k)^2 + m_2^2) = \\ &= -k^2(\alpha_1 + \alpha_2) - 2\alpha_2 pk + \alpha_1 m_1^2 + \alpha_2 m_2^2 - \alpha_2 p^2 \end{aligned} \quad (5.34)$$

Полиномы Симанзика:

$$U = \alpha_1 + \alpha_2 = 1 \quad (5.35)$$

$$F = \frac{1}{\mu^2} \left[ (\alpha_2 p)^2 + \alpha_1^2 m_1^2 + \alpha_2^2 m_2^2 - \alpha_2 p^2 \right] (\alpha_1 + \alpha_2) \quad (5.36)$$

В фейнмановском представлении  $1 = \alpha_1 + \alpha_2$ . Тогда

$$p^2 (\alpha_2^2 - \alpha_2) = -p^2 \alpha_1 \alpha_2 \quad (5.37)$$

В случае, когда массы равны нулю  $m_1 = m_2 = 0$

$$F = \alpha_1 \alpha_2 \left( -\frac{p^2}{\mu^2} \right) \quad (5.38)$$

И тогда интеграл

$$I_{\nu_1 \nu_2} = \frac{e^{\varepsilon \gamma_E} \Gamma(\nu - \frac{D}{2})}{\Gamma(\nu_1) \Gamma(\nu_2)} \cdot \left( -\frac{p^2}{\mu^2} \right)^{D/2 - \nu} \int_0^1 d\alpha \alpha^{D/2 - \nu_2 - 1} \cdot (1 - \alpha)^{D/2 - \nu_1 - 1} \quad (5.39)$$

В итоге,

$$I_{\nu_1 \nu_2} = e^{\varepsilon \gamma_E} \cdot \frac{\Gamma(\nu - \frac{D}{2}) \Gamma(\frac{D}{2} - \nu_1) \Gamma(\frac{D}{2} - \nu_2)}{\Gamma(\nu_1) \Gamma(\nu_2) \Gamma(D - \nu)} \left( \frac{p^2}{\mu^2} \right)^{D/2 - \nu} \quad (5.40)$$

Можно убедиться, что в случае  $\nu_1 = \nu_2 = 1$ , раскладывая  $\Gamma$ -функцию, получим:

$$I_{11} = \frac{1}{\varepsilon} + 1 - L + o(\varepsilon) \quad (5.41)$$

## Лекция 6. IBP reduction. Интегрирование по частям интегралов Фейнмана

### Тензорные структуры в интегралах

Интегралы Фейнмана могут включать в себя и тензорные структуры с несвернутыми индексами. Возникает вопрос, как в такой ситуации действовать. Рассмотрим пример

$$B^{\mu_1}(p, m_1, m_2) = e^{\varepsilon\gamma_E} \mu^{2\varepsilon} \int \frac{d^D k}{i\pi^{D/2}} \frac{k^{\mu_1}}{(-k_1^2 + m_1^2) \left( -(k-p)^2 + m_2^2 \right)} \quad (6.1)$$

этот интеграл пропорционален 4-вектору.

*Замечание:* Представленный метод будет работать только для однопетлевых интегралов.

Введем обозначения

$$B^{\mu_1} = p^{\mu_1} B_1 \quad (6.2)$$

и задача сводится к нахождению коэффициента  $B_1$ . Применим прием — метод Посорино-Велтмана. Заметим, что  $p_{\mu_1} k^{\mu_1}$  можно представить как

$$p_{\mu_1} k^{\mu_1} = \frac{1}{2} \left( -(k-p)^2 + m_2^2 \right) + \frac{1}{2} k^2 - \frac{1}{2} (-k^2 + m_1^2) + \frac{1}{2} m_1^2 - \frac{1}{2} m_2^2 \quad (6.3)$$

Умножим (6.3) на  $p^{\mu_1}$ , применяя формулу выше:

$$p^{\mu_1} B^{\mu_1} = p^2 B_1 = e^{\varepsilon\gamma_E} \mu^{2\varepsilon} \frac{1}{2} \int \frac{d^D k}{i\pi^{D/2}} \frac{\left( -(k-p)^2 + m_2^2 \right) + p^2 - (-k^2 + m_1^2) + m_1^2 - m_2^2}{(-k^2 + m_1^2) \left( -(k-p)^2 + m_2^2 \right)}. \quad (6.4)$$

Разумно ввести еще 2 интеграла:

$$B_0 = e^{\varepsilon\gamma_E} \mu^{2\varepsilon} \int \frac{d^D k}{i\pi^{D/2}} \frac{1}{(-k^2 + m_1^2) \left( -(k-p)^2 + m_2^2 \right)} \quad (6.5)$$

$$A_0(m_1) = e^{\varepsilon\gamma_E} \mu^{2\varepsilon} \int \frac{d^D k}{i\pi^{D/2}} \frac{1}{(-k^2 + m_1^2)}$$

И после замены  $k+p \rightarrow k$ :

$$p^{\mu_1} B^{\mu_1} = \frac{1}{2} A_0(m_1) + \frac{1}{2} (p^2 + m_1^2 - m_2^2) B_0 - \frac{1}{2} A_0(m_2) \quad (6.6)$$

Чтобы найти искомый  $B_1$  необходимо это выражение просто поделить на  $p^2$ .

Аналогичную процедуру можно провести для структур с двумя индексами, например, если

$$B^{\mu_1 \mu_2}(p, m_1, m_2) = e^{\varepsilon\gamma_E} \mu^{2\varepsilon} \int \frac{d^D k}{i\pi^{D/2}} \frac{k^{\mu_1} k^{\mu_2}}{(-k_1^2 + m_1^2) \left( -(k-p)^2 + m_2^2 \right)}. \quad (6.7)$$

Отличие состоит в том, что ситуация окажется «двойкая» — искомый интеграл будет представляться в виде суммы двух структур с двумя индексами:

$$B^{\mu_1\mu_2} = p^{\mu_1} p^{\mu_2} B_{11} + g^{\mu_1\mu_2} B_{12} \quad (6.8)$$

Видим, что все тензорные структуры в случае одной петли можно выразить через скалярные произведения — они являются приводимыми. В многопетлевом случае, вообще говоря, это не так.

## Метод интегрирования по частям

У этого метода есть авторы. Литература по теме

- *Chetyrkin, Tkachov, 1981, Nucl Phys* — метод для вычисления 4-петлевой  $\beta$ -функции (*русская многопетлевая революция*).

В англоязычной литературе этот метод называется *IBP-reduction* (Integration by parts).

Рассмотрим интеграл вида

$$F(a) = \int \frac{d^D k}{(k^2 - m^2)^a}. \quad (6.9)$$

Заметим, что если рассмотреть интеграл другого вида:

$$\int \frac{\partial}{\partial k_\mu} \left( \frac{k_\mu}{(k^2 - m^2)^a} \right) d^D k = 0, \quad (6.10)$$

этот интеграл равен нулю по теореме Гаусса. С другой стороны, можно вычислить производную явно:

$$\begin{aligned} \frac{\partial}{\partial k_\mu} \left( \frac{k_\mu}{(k^2 - m^2)^a} \right) &= \frac{\partial k_\mu}{\partial k_\mu} \frac{1}{(k^2 - m^2)^a} + k_\mu \frac{\partial}{\partial k_\mu} \frac{1}{(k^2 - m^2)^a} = \\ &= D \frac{1}{(k^2 - m^2)^a} - k_\mu \frac{a \cdot 2k_\mu}{(k^2 - m^2)^{a+1}} = \frac{D}{(k^2 - m^2)^a} - 2 \frac{ak^2}{(k^2 - m^2)^{a+1}} = \\ &= \frac{D - 2a}{(k^2 - m^2)^a} - \frac{2am^2}{(k^2 - m^2)^{a+1}}. \end{aligned} \quad (6.11)$$

Тогда в наших терминах:

$$(D - 2a)F(a) - 2am^2 F(a + 1) = 0 \quad (6.12)$$

Хотим получить рекуррентную формулу, для чего превратим  $a \rightarrow a - 1$ :

$$(D - 2a + 2)F(a - 1) - 2(a - 1)m^2 F(a) = 0 \quad (6.13)$$

Тогда

$$F(a) = \frac{D-2a+2}{2(a-1)m^2} F(a-1) \quad (6.14)$$

получили рекуррентную формулу — связали член в каждом порядке с членом в предыдущем. Можем итеративно проводить эту процедуру, находя  $F(a-1)$ , пока не дойдем до  $a=1$  ( $F(0)$  вычислить уже не получится — интеграл расходится):

$$I_1 = \int \frac{d^D k}{k^2 - m^2} = -i\pi^{D/2} \Gamma\left(1 - \frac{D}{2}\right) (m^2)^{D/2-1} \quad (6.15)$$

Таким образом,

$$F(a) = \frac{(-1)^a \left(1 - \frac{D}{2}\right)_{a-1}}{(a-1)! \cdot (m^2)^{a-1}} I_1 \quad (6.16)$$

где конструкция с нижним индексом  $\left(1 - \frac{d}{2}\right)_{a-1}$  — т.н. символ Похгаммера (убывающий/возрастающий факториал):

$$(x)_n = x(x+1)(x+2)\dots(x+(n-1)) \quad (6.17)$$

Последним элементом символа  $\left(1 - \frac{D}{2}\right)_{a-1}$  является

$$1 - \frac{D}{2} + a - 1 - 1 = \frac{1}{2}(-1)(D-2a+2), \quad (6.18)$$

поэтому мы учли в формуле множитель  $(-1)^a$ .

Таким образом, мы свели бесконечное семейство интегралов к одному интегралу. Интеграл, к которому мы сводим другие интегралы называется  $+^*$  *мастер-интегралом*.

- 1) Есть такие мастер-интегралы, которые больше ни к чему не сводятся — *истинные мастер-интегралы*
- 2) Есть такие, которые можно к чему-то свести, но иногда это не самый легкий путь.

Рассмотрим еще один интеграл

$$F(a_1, a_2) = \int \frac{d^D k}{(k^2)^{a_1} [(q-k)^2]^{a_2}}, \quad (6.19)$$

который соответствует диаграмме на рисунке 6.1. Здесь надо рассмотреть интегрирование по частям интеграла вида:

$$\int d^D k \frac{\partial}{\partial k_\mu} \left( \frac{k_\mu}{(k^2)^{a_1} ((q-k)^2)^{a_2}} \right) = 0 \quad (6.20)$$

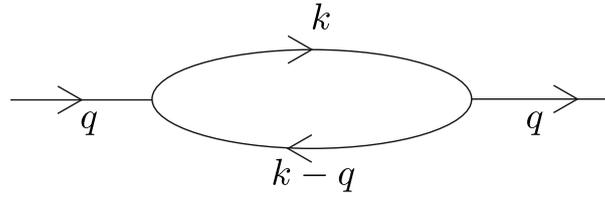


Рис. 6.1. Диаграмма, соответствующая интегралу  $F(a_1, a_2)$

и рассмотрим отдельно результат дифференцирования подынтегрального выражения

$$\begin{aligned} \frac{\partial}{\partial k_\mu} \left( \frac{k_\mu}{(k^2)^{a_1} ((q-k)^2)^{a_2}} \right) &= \frac{D}{(k^2)^{a_1} [(q-k)^2]^{a_2}} - \frac{2k_\mu k_\mu a_1}{(k^2)^{a_1+1} ((q-k)^2)^{a_2}} - \\ &- \frac{2(k-q)_\mu k_\mu a_2}{((q-k)^2)^{a_2+1} (k^2)^{a_1}} = \frac{D - 2a_1 - a_2}{(k^2)^{a_1} ((q-k)^2)^{a_2}} - \frac{a_2}{((q-k)^2)^{a_2+1} (k^2)^{a_1-1}} + \frac{a_2 q^2}{((q-k)^2)^{a_2+1} (k^2)^{a_1}} \end{aligned} \quad (6.21)$$

В последнем преобразовании было использовано соотношение:

$$2(k-q)_\mu k_\mu = 2k^2 - 2k_\mu q_\mu = 2k^2 + (k^2 - q^2)^2 - k^2 - q^2 = k^2 - q^2 + (k-q)^2 \quad (6.22)$$

Таким образом, получаем:

$$[D - 2a_1 - a_2 - a_2 2^+(1^- - q^2)] F(a_1, a_2) = 0, \quad (6.23)$$

где операторы  $2^+$  и  $1^-$ :

$$2^+ F(a_1, a_2) = F(a_1, a_2 + 1) \quad (6.24)$$

$$1^- F(a_1, a_2) = F(a_1 - 1, a_2) \quad (6.25)$$

Можем записать рекуррентные соотношения

$$(D - 2a_1 - a_2) F(a_1, a_2) - a_2 F(a_1 - 1, a_2 + 1) + a_2 q^2 F(a_1, a_2 + 1) = 0 \quad (6.26)$$

и после замены  $a_2 \rightarrow a_2 - 1$ :

$$(D - 2a_1 - a_2 + 1) F(a_1, a_2 - 1) - (a_2 - 1) F(a_1 - 1, a_2) + (a_2 - 1) q^2 F(a_1, a_2) = 0 \quad (6.27)$$

Откуда

$$F(a_1, a_2) = -\frac{1}{(a_2 - 1)q^2} [(D - 2a_1 - a_2 + 1) F(a_1, a_2 - 1) - (a_2 - 1) F(a_1 - 1, a_2)] \quad (6.28)$$

Можно сказать, что при вычислении интегралов мы идем по «сетке» из  $a_2$  и  $a_1$  (см. рис. 6.2) Чтобы сдвигаться не только по оси  $a_2$ , но и по оси  $a_1$ , необходимо применить

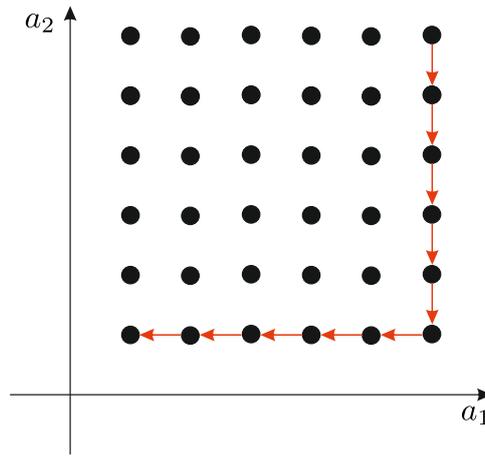


Рис. 6.2. Сетка параметров  $a_1$  и  $a_2$

соображения симметрии. Можно заметить, что интеграл симметричен относительно перестановки  $a_1 \leftrightarrow a_2$ .

$$F(a_2, a_1) = \int \frac{d^D k}{(k^2)^{a_2} [(q-k)^2]^{a_1}} = \left| k \rightarrow k+q, k \rightarrow -k \right| = \frac{d^D k}{(k^2)^{a_1} [(q-k)^2]^{a_2}} \quad (6.29)$$

Таким образом

$$F(a_2, a_1) = F(a_1, a_2) = \frac{1}{(a_1 - 1)q^2} [(D - 2a_2 - a_1 + 1)F(a_2, a_1 - 1) - (a_1 - 1)F(a_2 - 1, a_1)] \quad (6.30)$$

Получилась система уравнений, откуда  $F(a_2, a_1 - 1)$  можно выразить через  $F(a_2 - 1, a_1)$ . То есть, сначала спустившись по  $a_2$  до  $a_2 = 1$ , потом можем дойти вдоль этой прямой до точки  $a_1 = a_2 = 1$ . Останется 1 неприводимый мастер-интеграл  $F(1, 1)$ .

Если у нас есть интеграл, даже многопетлевой вида, можем написать для любого петлевого импульса:

$$\int d^D k d^D l \dots f(k, l \dots) \int \frac{\partial}{\partial k_\mu} (k_\mu f) = 0 \quad (6.31)$$

Если у нас есть внешний импульс  $p_\mu$ , также можем написать:

$$\int d^D k d^D l \dots f(k, l \dots) \int \frac{\partial}{\partial k_\mu} (p_\mu f) = 0 \quad (6.32)$$

## Использование Лоренц-инвариантности

Интегралы, которые мы рассматриваем, являются скалярами относительно группы Лоренца. Тогда генератор группы Лоренца при действии на интегралы должен давать ноль (*LI relation*):

$$\sum_{i=1}^{n_{\text{ext}}} \left( p_i^\nu \frac{\partial}{\partial p_{\mu i}} - p_i^\mu \frac{\partial}{\partial p_{\nu i}} \right) I(\dots) = 0 \quad (6.33)$$

## Безмасштабные интегралы (Scaleless)

Рассмотрим интеграл

$$\int \frac{d^D k}{i\pi^{D/2}} (-k^2)^a \quad (6.34)$$

Его размерность  $2a + D$ . Но по своему построению, у этого интеграла нет характерного масштаба (нет массы, внешнего импульса,  $\mu$ ). В размерной технике такой интеграл равен нулю.

Когда мы интегрируем по частям, можно сразу избавляться от безмасштабных интегралов.

Также безмасштабным являются интегралы вида:

$$\int \frac{d^D l_1 d^D l_2}{(l_1 - p)^2 l_2^2 (l_1 - p - l_2)^2} \quad (6.35)$$

С одной стороны, в этот интеграл входит внешний импульс  $p$ , но его можно исключить заменой  $l_1 \rightarrow l_1 + p$ . Значит, в размерной технике этот интеграл также равен нулю.

Если есть преобразование

$$l_1 \rightarrow \alpha l_1 + (1 - \alpha)p \quad (6.36)$$

$$l_2 \rightarrow \alpha l_2 \quad (6.37)$$

Тогда из под интеграла выходит коэффициент  $\alpha^{\dots}$ , а сам интеграл остается тем же, но без внешнего импульса.

На последнем шаге интегрирования по частям имеет смысл проверять при помощи таких преобразований равенство нулю мастер-интегралов.

## Угловой интеграл

Рассмотрим интеграл вида (6.19). Введем обозначение  $\theta_\alpha = \theta(a_\alpha - \frac{1}{2})$  — функция Хэвисайда. Принято семейство интегралов с соответствующими степенями в знаменателе записывать через набор  $\vec{\theta} = \{\theta_1, \theta_2, \dots, \theta_n\}$ . Принято говорить, что у такого семейства есть *угловой интеграл*  $I_{\theta_1, \theta_2, \theta_3, \theta_4, \dots}$  (интеграл со степенями равными единице в знаменателе, к которому сводятся интегралы после *IBP reduction*).

**Утв.** Если угловой интеграл равен нулю, то все остальные интегралы также равны нулю.

Это утверждение часто применялось при рассмотрении диаграмм Фейнмана. Однако, проблема состоит в том, что существуют интегралы, для которых это не так.

Пусть  $k$  — внешний импульс:

$$F(a_1, a_2) = \int \frac{d^D l}{(l^2)^{a_1} ((l - k)^2)^{a_2}} \quad (6.38)$$

Этот интеграл сводится к  $I_{11} \sim I_{20}$ , что не сводится к нулю. Покажем, что это действительно так. Введем произвольный импульс  $\tilde{k}^2 = 0$ ,  $(\tilde{k} \cdot k) = 1$  и подействуем на интеграл оператором дифференцирования:

$$\frac{\partial}{\partial l_\mu} ((l_\mu + (l \cdot k) \tilde{k}_\mu - (l \cdot \tilde{k}) k_\mu) F(1, 1)) = (D - 4)F(1, 1) \quad (6.39)$$

$$\begin{aligned} D + k_\mu \tilde{k}_\mu - \tilde{k}_\mu k_\mu - \left( \frac{2l_\mu}{(l^2)^2 (l-k)^2} + \frac{2(l_\mu - k_\mu)}{l^2 ((l-k)^2)^2} \right) (l_\mu + (l \cdot k) \tilde{k}_\mu - (l \cdot \tilde{k}) k_\mu) = \quad (6.40) \\ = D - \frac{2}{l^2 (l-k)^2} - \frac{2}{((l-k)^2)^2} + \frac{(l-k)^2 - l^2 - k^2}{l^2 ((l-k)^2)^2} - \frac{2l \cdot k}{l^2 (l^2 - k^2)^2} = \\ = D \frac{1}{(k^2)^2 (l-k)^2} - 2 \frac{l^2 (l-k)^2 + l^4 - 2lk l^2 + k^2 l^2}{(l^2)^2 ((l-k)^2)^2} \end{aligned}$$

Получаем исходный интеграл с коэффициентом  $-4$ . *Альтернативный подход был предложен Романом Ли из Новосибирска в 2008 году.*

# Лекция 7. Безмасштабные интегралы. Критерий Ли.

## Алгоритм Лапорта

### Идея метода IBP reduction

Метод основан на том, что интеграл вида

$$\int \frac{d^D k}{(2\pi)^D} \cdots \frac{d^D k_n}{(2\pi)^D} \frac{\partial}{\partial k^\mu} \left( \frac{k_\mu}{k^2 ((k-p)^2 - m^2) (\dots)} \right) = 0 \quad (7.1)$$

равен нулю по теореме Гаусса. Этот результат не зависит от количества петлевых интегралов, производных по петлевым импульсам. Кроме того, дифференцировать можно как по всем петлевым импульсам, так и по внешним.

На основании этой формулы можно получить соотношения между интегралами такого вида.

Введем обозначение для интегралов:

$$I(v_1, v_2, v_3, \dots) \equiv \int \frac{d^D k}{(2\pi)^D} \cdots \frac{d^D k_n}{(2\pi)^D} \left( \frac{k_\mu}{(k^2)^{v_1} ((k-p)^2 - m^2)^{v_2} (\dots)^{v_3}} \right). \quad (7.2)$$

Такие интегралы могут быть выражены через интегралы с меньшими степенями:

$$I_{v_1 v_2 v_3 \dots} = \sum_i C_i (I_i)_{v_1 - x_1, v_2 - x_2, \dots} \quad (7.3)$$

Идея заключается в том, что интегралы такого вида можно свести к простейшим интегралам, которые называются *мастер-интегралами*.

В прошлый раз мы рассмотрели несколько примеров. Так, следующий интеграл сводится к интегралу, у которого знаменатель в первой степени  $v_1 = 1$ .

$$\int \frac{d^D k}{(k^2 - m^2)^{v_1}} \rightarrow I_1 \quad (7.4)$$

Таким образом, формируется целое семейство интегралов, которое называется *сектором* и записывается как

$$\theta_i = \theta \left( v_i - \frac{1}{2} \right) - \text{функция Хевисайда} \quad (7.5)$$

И интегралы приводятся к семейству интегралов в пространстве векторов (или в секторе)  $\vec{\theta} = (\theta_1, \theta_2, \dots) - I_{\theta_1 \theta_2 \theta_3 \dots}$ .

## Безмасштабные интегралы

Существует класс интегралов, которые являются *безмасштабными*, и в размерной технике всегда равны нулю:

$$\int \frac{d^D k}{(2\pi)^D} \frac{1}{k^2} = 0 \quad (7.6)$$

Конечно, важно проводить учет таких интегралов в рамках реализации IBP reduction на каждом шаге, т.к. это позволяет значительно снизить вычислительные ресурсы.

Критерий безмасштабности интеграла — *если существует такое преобразование, которое бы сводило интеграл к самому себе с некоторым коэффициентом, то интеграл является безмасштабным.*

В качестве примера, рассмотрим интеграл вида

$$\int \frac{d^D l_1 d^D l_2}{l_2^2 (l_1 - p)^2 (l_1 - p + l_2)^2}. \quad (7.7)$$

Тогда при замене  $l_1 - p \rightarrow l_1$  интеграл становится явно безмасштабным.

В общем виде преобразование выглядит как

$$l_1 \rightarrow \alpha l_1 + (1 - \alpha)p \quad (7.8)$$

$$l_2 \rightarrow \alpha l_2 \quad (7.9)$$

Тогда безмасштабный интеграл преобразуется как

$$I \rightarrow \alpha^{2D-6} I. \quad (7.10)$$

Подбирать вид преобразования для каждого конкретного интеграла — трудоемкая задача. Хочется сформулировать критерий, который бы не зависел от конкретного вида преобразования.

Долгое время использовался эмпирический критерий — *Если угловой интеграл (интеграл на последнем шаге итерации) равен нулю, то и все семейство интегралов, сводящихся к нему, равняется нулю.* Эта идея, несмотря на свою простоту, работала довольно неплохо — для большого числа интегралов это оказывается верно. Однако можно найти пример, когда этот критерий не работает.

Рассмотрим семейство интегралов безмасштабных на массовой поверхности  $p^2 = 0$ :

$$I_{v_1 v_2} = \int \frac{d^D k}{(k^2)^{v_1} ((k-p)^2)^{v_2}}. \quad (7.11)$$

Однако угловой интеграл, к которому он сводится  $I_{v_1 v_2} \rightarrow I_{11}$  не равен нулю. Чтобы он стал равен нулю в результате IBP необходимо провести преобразование.

Рассмотрим преобразование

$$\hat{O} = \partial_k \left( k + (kp) \tilde{p} + (k\tilde{p}) p \right), \quad (7.12)$$

где  $\tilde{p}^2 = 0$  и  $(\tilde{p} \cdot p) = 1$ . При действии такого преобразования на угловой интеграл  $I_{11}$ :

$$\hat{O}I_{11} = (4 - D)I_{11} \quad (7.13)$$

видим, что интеграл является безмасштабным.

Преобразование  $\hat{O}$  является генератором преобразований

$$k_\mu \rightarrow k_\mu + \varepsilon \left( k_\mu + (kp) \tilde{p}_\mu + (k\tilde{p}) p_\mu \right) \quad (7.14)$$

*Критерий был сформулирован Романом Ли из Новосибирска в 2013 году.*

- *Lee, Pomerskiy, JHEP, 2013*

Введем еще один способ параметризации.

## Представление Ли-Померанского

$$I = \frac{e^{l\varepsilon\gamma_E} \Gamma(D/2)}{\Gamma\left(\frac{(l-1)D}{2} - \nu\right) \prod_{j=1}^{n_{ext}} \Gamma(\nu_j)} \int_{u_j > 0} d^{n_{int}} u \left( \prod_{j=1}^{n_{int}} u_j^{\nu_j - 1} \right) [G(u)]^{-p/2}, \quad (7.15)$$

где  $\nu = \sum_i \nu_i$ ,  $G(u) = U(u) + F(u)$ ,  $U$  и  $F$  — полиномы Симанзика.

Перейдем от представления Ли-Померанского к представлению Фейнмана, чтобы проверить справедливость формулы (7.15). Будем использовать следующее полезное соотношение:

$$1 = \int_0^{+\infty} dt \delta\left(t - \sum_{j=1}^{n_{int}} u_j\right). \quad (7.16)$$

Введем обозначение:

$$a_j = \frac{u_j}{t} \quad (7.17)$$

и вспомним, что  $U$  однородный полином по  $l$ , а  $F$  однородный полином по  $l + 1$  (5.24).

$$I = \frac{e^{l\varepsilon\gamma_E} \Gamma(D/2)}{\Gamma\left(\frac{(l-1)D}{2} - \nu\right) \prod_{j=1}^{n_{ext}} \Gamma(\nu_j)} \int_{a_j > 0} d^{n_{int}} a \delta\left(1 - \sum_{j=1}^{n_{int}} a_j\right) \left(\prod_{j=1}^n a_j^{\nu_j - 1}\right) \times \\ \times \int dt t^{\nu - lD/2 - 1} [U(a) + F(a)t]^{-D/2} \quad (7.18)$$

Произведем замену  $t \rightarrow tU(a)F(a)$ . Тогда получается, что интеграл переписывается как

$$I = \frac{e^{l\epsilon\gamma_E} \Gamma(D/2)}{\Gamma\left(\frac{(l-1)D}{2} - \nu\right) \prod_{j=1}^{n_{ext}} \Gamma(\nu_j)} \int_{a_j > 0} d^{n_{int}} a \delta\left(1 - \sum_{j=1}^{n_{int}} a_j^{\nu_j-1}\right) \left(\prod_{j=1}^{n_{int}} a_j^{\nu_j-1}\right) \times \\ \times \int_0^{+\infty} dt t^{\nu-lD/2-1} [1+t]^{-D/2} \frac{U^{\nu-\frac{l+1}{2}D}}{F^{\nu-\frac{ld}{2}}} \quad (7.19)$$

заметим, что

$$\int_0^{+\infty} dt t^{\nu-lD/2-1} [1+t]^{-D/2} = \frac{\Gamma(\nu - \frac{lD}{2}) \Gamma\left(\frac{(l+1)D}{2} - \nu\right)}{\Gamma\left(\frac{D}{2}\right)} \quad (7.20)$$

и получаем интеграл в Фейнмановском представлении:

$$I = \frac{e^{l\epsilon\gamma_E}}{\prod_{j=1}^{n_{ext}} \Gamma(\nu_j)} \int_{a_j > 0} d^{n_{int}} a \delta\left(1 - \sum_{j=1}^{n_{int}} a_j^{\nu_j-1}\right) \left(\prod_{j=1}^{n_{int}} a_j^{\nu_j-1}\right) \times \frac{U^{\nu-\frac{l+1}{2}D}}{F^{\nu-\frac{ld}{2}}}. \quad (7.21)$$

Таким образом, мы доказали справедливость представления Ли-Померанского.

## Критерий Ли

Предположим что существует инфинитезимальное преобразование вида

$$u_j \rightarrow \tilde{u}_j = (1 + k_j \omega) u_j \quad (7.22)$$

такое, чтобы функция  $G$  преобразовывалась как

$$G(\tilde{u}) = (1 + \omega) G(u). \quad (7.23)$$

Тогда интеграл  $I$  в представлении Ли-Померанского (7.15) перейдет в:

$$I \rightarrow I' = \left[1 + \omega \left(\sum_j k_j - \frac{D}{2}\right)\right] I \quad (7.24)$$

Если удастся найти такие  $k_j$ , то интеграл безмасштабный (интеграл в результате преобразования равен самому себе с некоторым коэффициентом).

Сформулируем **условие Ли**, преобразуя формулу (7.23)

Если существуют такие  $k_j$ , что при преобразовании (7.22) выполняется

$$\frac{G(\tilde{u}) - G(u)}{\omega} = G(u), \quad (7.25)$$

или в пределе:

$$\boxed{\sum_i k_i u_i \frac{\partial G}{\partial u_i} = G} \text{ — критерий Ли (2013)} \quad (7.26)$$

то интеграл (7.15) является безмасштабным. Если таких  $k_j$  не существует, то интеграл не является безмасштабным.

## Пример применения критерия Ли

Рассмотрим интеграл

$$I_{11} = \int \frac{d^D k}{k^2 (k-p)^2} \quad (7.27)$$

Для этого интеграла мы уже получали полиномы Симанзика (см. 5.36)

$$U = a_1 + a_2, \quad F = a_1 a_2 \left( -\frac{p^2}{\mu^2} \right) \quad (7.28)$$

тогда

$$G = u_1 + u_2 + \underbrace{u_1 u_2}_{\equiv x} \left( -\frac{p^2}{\mu^2} \right) \quad (7.29)$$

Применим критерий

$$k_1 u_1 (1 + x u_2) + k_2 u_2 (1 + x u_1) = u_1 + u_2 + x (u_1 u_2), \quad (7.30)$$

тогда получаем

$$k_1 u_1 + k_2 u_2 + (k_1 + k_2) x u_1 u_2 = u_1 + u_2 + x u_1 u_2 \quad (7.31)$$

и наша система:

$$\begin{cases} k_1 = 1 \\ k_2 = 1 \\ k_1 + k_2 = 1 \end{cases} \quad (7.32)$$

Можно заметить, что система несовместна, а значит интеграл не безмасштабный.

В случае же, когда мы находимся на массовой оболочке  $p^2 = 0 \implies x = 0$ , система имеет решение

$$\begin{cases} k_1 = 1 \\ k_2 = 1 \end{cases} \quad (7.33)$$

и интеграл является безмасштабным.

## Алгоритм Лапорта

Иногда, гораздо проще вычислить не угловые интегралы (интегралы на последнем шаге IBP reduction), а какой-то интеграл «перед ним». Для того, чтобы понять, на каком шаге остановиться, необходимо найти критерий «сложности» интеграла. Алгоритм Лапорта — эмпирическая оценка сложности интегралов.

- Laporta, 2000

Лапорта предложил рассматривать несколько величин:

- 1)  $N_{\text{prop}} = \sum_{j=1}^{n_{\text{int}}} \theta \left( v_j - \frac{1}{2} \right)$  — число пропагаторов;
- 2) Идентификатор сектора. Сектор определяется вектором  $\vec{\theta} = (\theta_1, \theta_2, \dots, \theta_n)$  или на языке компьютера — последовательностью 0 и 1.

$$N_{\text{id}} = \sum_{j=1}^{n_{\text{int}}} 2^{j-1} \times \theta \left( v_j - \frac{1}{2} \right) \quad (7.34)$$

- 3) Сумма всех положительных степеней  $v_j$  в знаменателе

$$r = \sum_{j=1}^{n_{\text{int}}} v_j \theta \left( v_j - \frac{1}{2} \right) \quad (7.35)$$

- 4) Сумма всех положительных степеней в числителе

$$s = \sum_{j=1}^{n_{\text{int}}} |v_j| \theta \left( \frac{1}{2} - v_j \right) \quad (7.36)$$

Каждому интегралу можно поставить в соответствие вектор:

$$(N_{\text{prop}}, N_{\text{id}}, r, s) \text{ — ISP-базис} \quad (7.37)$$

и можем сравнивать эти вектора в лексикографическом порядке — чем больше величина, тем сложнее интеграл. При таком порядке величин в числителе самого маленького интеграла могут остаться  $p_\mu q^\mu$ . Можно рассматривать эту величину как эмпирическую оценку сложности интеграла.

Иногда используют также

$$(N_{\text{prop}}, N_{\text{id}}, s, r) \text{ — точечный базис.} \quad (7.38)$$

В таком базисе в знаменателе интеграла могут остаться множители в степенях, не равных 1.

### Общий алгоритм нахождения мастер-интегралов:

- 1) Выписываем соотношения интегрирования по частям — *IBP reduction*, соотношения Лоренц-инвариантности, симметрии системы.
- 2) Решаем систему уравнений. Находим, как связаны более простые интегралы с предыдущими.
- 3) Алгоритм Лапорта + отсев безмасштабных интегралов.

## Лекция 8. Программы LiteRed и Kira

В прошлый раз мы разобрали метод интегрирования по частям. На этой лекции будет продемонстрирована работа двух программ, реализующие это технически. Проблема заключается в том, что большинство программ, решающих эту задачу, работают только для UNIX систем.

### Программа LiteRed

Эту программу создал Роман Ли. Исторически она не самая первая, однако работа в ней интуитивно понятна. Ссылка на статью:

- [arXiv:1310.1145](https://arxiv.org/abs/1310.1145)

### Установка программы LiteRed

- 1) Перейти на сайт <https://www.inp.nsk.su/~lee/programs/LiteRed/> и установить программу во вкладке Download. Можно также сразу перейти на GitHub проекта.
- 2) Копируем все файлы папки Source в произвольную папку.
- 3) Открыть в Mathematica скрипт `makeShortcut.m`.
- 4) Для работы достаточно запустить Mathematica, и инициализировать библиотеку командами

```
<< LiteRed2';  
SetDirectory[NotebookDirectory[]];
```

### Пример работы в программе LiteRed

Рассмотрим пример однопетлевой безмассовой вершины (см. рис. 8.1).

- 1) Работу с этим пакетом всегда начинаем с его инициализации, как было показано в последнем пункте предыдущего раздела.
- 2) Задаем размерность и набор переменных (в нашем случае импульсы  $p$  и  $q$ ).

```
SetDim[d];  
Declare[{l, p, q}, Vector];
```

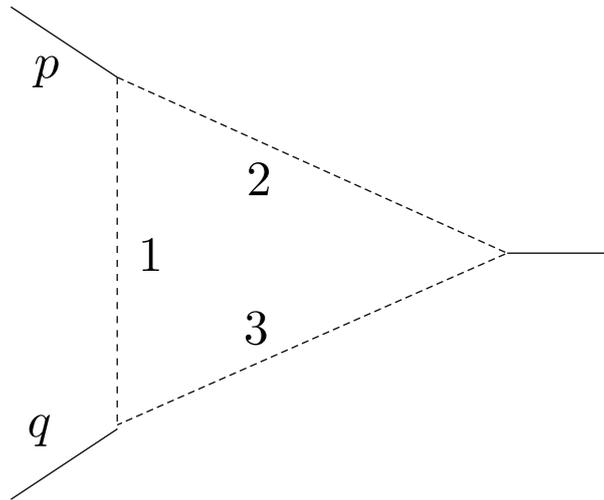


Рис. 8.1. Однопетлевая безмассовая вершина

- 3) Определяем базис и запускаем поиск симметрий и правил редукции

```
NewBasis[v1, {1, p - 1, q + 1}, {1}, Directory -> "v1 dir"]
GenerateIBP[v1]
AnalyzeSectors[v1]
FindSymmetries[v1]
```

- 4) Сохраняем результат, и выходим из ядра

```
DiskSave[v1]
Quit[]
```

- 5) Снова загружаем библиотеку, базис и запускаем редукцию

```
<< LiteRed2';
SetDirectory[NotebookDirectory[]];
SetDim[d];
Declare[{1, p, q}, Vector];
<< "v1 dir/v1";

IBPReduce[j[v1, 1, 2, 3] + j[v1, 3, 2, 1]]
```

- 6) Упрощаем результат

```
Collectj[IBPReduce[
Dinv[j[v1, 1, 1, 1], p\[CenterDot]q]] /. {p\[CenterDot]p ->
Subscript[m, 1]^2, q\[CenterDot]q -> Subscript[m, 2]^2,
p\[CenterDot]q -> (Subscript[m, 3]^2 - Subscript[m, 1]^2 -
Subscript[m, 2]^2)/2}, Factor]
```

7) Вывод программы имеет вид, представленный на рис. 8.2.

$$\text{Out}[7]= \frac{4(-3+d)j[v1, 0, 1, 1]}{(m_1 - m_2 - m_3)(m_1 + m_2 - m_3)(m_1 - m_2 + m_3)(m_1 + m_2 + m_3)} + \frac{2(-3+d)j[v1, 1, 1, 0](m_1^2 - m_2^2 + m_3^2)}{m_3^2(-m_1 - m_2 + m_3)(m_1 - m_2 + m_3)(-m_1 + m_2 + m_3)(m_1 + m_2 + m_3)} + \frac{2(-3+d)j[v1, 1, 0, 1](-m_1^2 + m_2^2 + m_3^2)}{m_3^2(-m_1 - m_2 + m_3)(m_1 - m_2 + m_3)(-m_1 + m_2 + m_3)(m_1 + m_2 + m_3)} - \frac{(j[v1, 1, 1, 1](4m_1^4 - d m_1^4 - 8 m_1^2 m_2^2 + 2 d m_1^2 m_2^2 + 4 m_2^4 - d m_2^4 - 2 m_1^2 m_3^2 - 2 m_2^2 m_3^2 - 2 m_3^4 + d m_3^4))}{(m_3^2(-m_1 - m_2 + m_3)(m_1 - m_2 + m_3)(-m_1 + m_2 + m_3)(m_1 + m_2 + m_3))}$$

Рис. 8.2. Результат редукции программы

### Дополнительные замечания по работе с программой LiteRed

1) Также можно задавать ограничения на переменные самостоятельно (например, чтобы учесть наличие безмасштабных интегралов). Для этого используется команда `SetConstraints`.

*Замчание:* Если задать такие ограничения, чтобы интеграл стал безмасштабным — то при редукции сразу выдаст 0. Программа умеет проверять *критерий Ли* 7.26.

- 2) Программа автоматически учитывает соотношения Лоренц-Инвариантности.
- 3) Вместе с программой поставляется обширная база примеров, чтобы посмотреть на работу программы в разных ситуациях. При работе со старыми примерами могут возникать проблемы — *не забудьте указать соответствующую версию при инициализации программы.*

## Программа Kira

Программа Kira является наиболее активно развивающимся проектом, посвященным интегрированию по частям интегралов Фейнмана, сведению сложных интегралов к более простым мастер-интегралам.

Важно отметить, что эта программа для LINUX-систем.

## Установка программы Kira

Стоит отметить, что устанавливается программа очень долго. В подробностях процесс установки описан не будет.

- 1) Программа требует загрузки и установки дополнительных библиотек, например `GiNaC`, `yaml-cpp`, `Fermat`. Их надо установить в первую очередь.
- 2) Программу можно скачать с GitHub'a проекта.
- 3) Далее — продолжить работать по инструкции с GitHub.

## Работа в программе Kira

- 1) Заходим в папку Kira
- 2) Можно ознакомиться с примерами вычислений в папке `examples`
- 3) Основной файл — `jobs.yaml`. В нем находится информация об идентификаторе сектора, максимальная сумма степеней пропагаторов, числителей и т.д.  
Синтаксис:

```
jobs:
- reduce_sectors:
reduce:
- {topologies: [T1,...], sectors: [S1,...],
r: rmax, s: smax, d: dmax}
- {...}
```

Здесь

- a) `rmax` — максимальная степень пропагаторов в знаменателе;
  - b) `smax` — максимальная степень в числителе;
  - c) `sectors` — номера сектора;
  - d) `topologies` — задаваемые топологии;
  - e) В том же файле можно задать предпочтительный вид мастер-интегралов, к которым сводятся интегралы Фейнмана.
- 4) Другим важным файлом является `integralfamilies.yaml`. Его характерный вид

```
integralfamilies:  
- name: "topo7"  
loop_momenta: [k1,k2]  
top_level_sectors: [127]  
propagators:  
- ["-k1", 0] #1  
- ["k2", 0] #2  
- ["-k1+k2", 0] #3  
- ["k1+q2", "m2^2"] #4  
- ["k2-p2", 0] #5  
- ["-k1+p1+p2", 0] #6  
- ["k2-p1-p2", 0] #7  
- ["k1-p2", 0] #8  
- ["k2-q2", 0] #9  
cut_propagators: [3,4]
```

В этом файле задаются пропагаторы, их массы.

- 5) Наконец, в файле `kinematics.yaml` можно задать кинематические соотношения.
- 6) Для запуска вычислений необходимо ввести в командной строке путь до бинарного исполняемого файла.
- 7) После выполнения программы в файле `masters` будут находиться получившиеся мастер-интегралы в виде матриц с цифрами.
- 8) Можно включить также алгебраическое представление мастер-интегралов. Для этого надо вставить код `algebraic_reconstruction: true` в соответствующий конфигурационный файл `jobs.yaml`.

Быстродействие этой программы действительно впечатляет, особенно если сравнивать с программами на базе `Mathematica`.

Для более глубокого знакомства с методами интегрирования по частям петлевых интегралов Фейнмана можно ознакомиться с книгой

- *Stefan Weinzierl — Feynman Integrals*

## Лекция 9. Численное вычисление интегралов Фейнмана. Метод Меллина-Барнса

В прошлый раз мы рассмотрели несколько программ, посвященных интегрированию по частям. В принципе, метод интегрирования по частям даёт возможность разложить интеграл Фейнмана по мастер-интегралам (некоему конечному классу интегралов).

Однако этот метод не дает возможности вычислить эти, даже простые, интегралы. И не так уж много существует программ, которые умеют решать такую задачу. Чаще всего программы решают эту задачу численно.

Достаточно большой класс интегралов можно взять численно и с большой точностью.

### Метод Меллина-Барнса

Меллин (Mellin) и Барнс (Barnes) — это два математика, жившие в 20 веке. Конечно, в то время об интегралах Фейнмана еще не было известно, однако метод назван так, потому что в нем используется формула, которая была получена этими математиками.

**Утв.**

$$\frac{1}{(A+B)^\lambda} = \frac{1}{2\pi i} \int_{-i\infty}^{+i\infty} \frac{\Gamma(\lambda-s)\Gamma(s)}{\Gamma(\lambda)} \frac{A^{s-\lambda}}{B^s} ds \quad (9.1)$$

Эту формулу можно назвать «обратной к формуле параметризации Фейнмана» в том смысле, что мы не сводим произведение к сумме под интегралом, а наоборот — сводим сумму к произведению. Так, например, характерный пропагатор сводится к

$$\frac{1}{(-k^2+m^2)^\lambda} = \frac{1}{2\pi i} \int_{-i\infty}^{i\infty} \frac{\Gamma(\lambda-s)\Gamma(s)}{\Gamma(\lambda)} \frac{(m^2)^{s-\lambda}}{(-k^2)^s} ds \quad (9.2)$$

Таким образом, интеграл, по сути, сводится к безмассовому, что значительно его упрощает. Далее этот интеграл берется с помощью вычетов.

Попробуем понять, откуда берется формула (9.1)

### Преобразование Меллина

Из курса ТФКП известно выражение для преобразования Меллина  $M$  от функции  $f$ :

$$\{Mf\}(s) = \varphi(s) = \int_0^{+\infty} x^{s-1} f(x) dx, \quad (9.3)$$

Можно совершить и обратное преобразование:

$$\{M^{-1}\varphi\}(x) = f(x) = \frac{1}{2\pi i} \int_{c-i\infty}^{c+i\infty} \lambda^{-s} \varphi(s) ds. \quad (9.4)$$

Вынесем в формуле (9.1)  $A$ :

$$\frac{1}{(A+B)^\lambda} = \frac{1}{A^\lambda \left(1 + \frac{B}{A}\right)^\lambda} = \frac{1}{A^\lambda (1+x)^\lambda}, \quad (9.5)$$

где  $x \equiv \frac{B}{A}$ . Совершим преобразование Меллина для этой функции:

$$\varphi(s) = \int_0^{+\infty} \frac{x^{s-1}}{(1+x)^\lambda} dx. \quad (9.6)$$

Вспомним определение Бета-функции:

$$B(x, y) = \int_0^\infty \frac{t^{x-1}}{(1+t)^{x+y}} dt \quad (9.7)$$

С другой стороны

$$B(x, y) = \frac{\Gamma(x)\Gamma(y)}{\Gamma(x+y)}, \quad (9.8)$$

И тогда

$$\varphi(s) = B(s, \lambda - s) = \frac{\Gamma(s)\Gamma(\lambda - s)}{\Gamma(\lambda)}. \quad (9.9)$$

Функцию  $f(x)$  можем выразить как

$$f(x) = \frac{1}{2\pi i} \int_{c-i\infty}^{c+i\infty} x^{-s} \frac{\Gamma(s)\Gamma(\lambda - s)}{\Gamma(\lambda)} ds. \quad (9.10)$$

И, наконец,

$$\frac{1}{\left(1 + \frac{B}{A}\right)^\lambda} = \frac{1}{2\pi i} \int_{c-i\infty}^{c+i\infty} \frac{\Gamma(s)\Gamma(\lambda - s)}{\Gamma(\lambda)} \left(\frac{B}{A}\right)^{-s} ds \quad (9.11)$$

Умножая последнее выражение на  $\frac{1}{A^\lambda}$ , получим

$$\frac{1}{(A+B)^\lambda} = \frac{1}{2\pi i} \int_{c-i\infty}^{c+i\infty} \frac{\Gamma(s)\Gamma(\lambda - s)}{\Gamma(\lambda)} \frac{A^{s-\lambda}}{B^s} ds \quad (9.12)$$

Таким образом, мы доказали формулу (9.1).

*Замечание:* Чаще всего эту формулу используют, чтобы находить значение интегралов с помощью представления Фейнмана.

В представлении Фейнмана возникают полиномы Симанзика  $U$  и  $F$ . Полином  $F$  находится в знаменателе, и он зависит от нескольких параметров. Идея состоит в том, чтобы с помощью формулы Меллина-Барнса брать интегралы по этим параметрам.

## Обобщение формулы Меллина-Барнса

Обсудим обобщение этой формулы. Применим эту формулу многократно. Рассмотрим интеграл, в котором  $B \rightarrow B + C$ :

$$\frac{1}{(A+B+C)^\lambda} = \frac{1}{2\pi i} \int_{-i\infty}^{+i\infty} \frac{\Gamma(\lambda-s)\Gamma(s)}{\Gamma(\lambda)} \frac{A^{s-\lambda}}{(B+C)^s} ds \quad (9.13)$$

После чего, применим ту же самую формулу для функции  $\frac{1}{(B+C)^s}$ . Таким образом, мы зарабатываем уже 2 интеграла в комплексной плоскости:

$$\frac{1}{(A+B+C)^\lambda} = \frac{1}{2\pi i} \int_{-i\infty}^{+i\infty} ds \frac{1}{2\pi i} \int_{-i\infty}^{+i\infty} dt \frac{\Gamma(\lambda-s)\Gamma(s)\Gamma(s-t)\Gamma(t)}{\Gamma(\lambda)} \frac{A^{s-\lambda} B^{t-s}}{C^t}. \quad (9.14)$$

В общем случае  $n$  членов в сумме в знаменателе, получим общий вид формулы:

$$\frac{1}{(A_1+A_2+\dots+A_n)^c} = \frac{1}{\Gamma(c)} \frac{1}{(2\pi i)^{n-1}} \int_{\gamma-i\infty}^{\gamma+i\infty} ds_1 \dots \int_{-i\infty}^{+i\infty} ds_{n-1} \times \\ \times \Gamma(-s_1) \dots \Gamma(-s_{n-1}) \Gamma(s_1 + \dots + s_{n-1} + c) A_1^{s_1} \dots A_{n-1}^{s_{n-1}} A_n^{-s_1 - \dots - s_{n-1} - c} \quad (9.15)$$

В этой формуле очень много  $\Gamma$ -функций. В общей формуле из этих функций «собирается» интеграл вида:

$$I = \frac{1}{2\pi i} \int ds \frac{\Gamma(s+a_1) \dots \Gamma(s+a_n) \Gamma(-s+b_1) \dots \Gamma(-s+b_n)}{\Gamma(s+c_1) \dots \Gamma(s+c_n) \Gamma(-s+d_1) \dots \Gamma(-s+d_n)} \quad (9.16)$$

Вспомним, как выглядит Гамма-функция с точки зрения контурного интегрирования. Для этого надо вспомнить, как расположены полюса у Гамма-функции (см. рис. 9.1). У Гамма-функции нет нулей, что часто позволяет игнорировать ее наличие в знаменателе, т.к. она не приводит к полюсам. В отрицательной же области Гамма-функция имеет *бесконечное число полюсов* в точках  $x = -1, -2, -3, \dots$

Тогда полюса в интеграле  $I$  обозначим  $a_i$  и они равны:

$$s = -a_n - n \quad (9.17)$$

Полюса по  $b_i$ :

$$s = b_n + n \quad (9.18)$$

Тогда контур интегрирования будем строить так, чтобы  $\gamma$  в формуле (9.15) удовлетворяла свойству

$$\max(a_n) < \gamma < \min(b_n) \quad (9.19)$$

В этом случае, интеграл можно брать просто вдоль мнимой оси — см. 9.2 случай А.

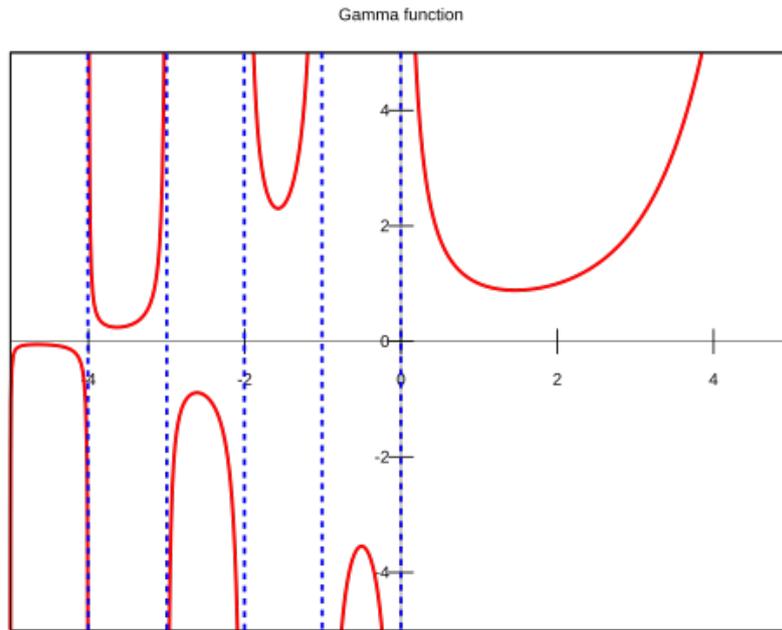


Рис. 9.1. График Гамма-функции

Представим ситуацию, когда  $b_1 < 0$  а  $a_1 < 0$ . Тогда получается, что полюса «на-  
 лезают» друг на друга, и необходимо деформировать контур таким образом, чтобы  
 все полюса, зависящие от  $\Gamma(-s)$  лежали справа от контура, а от  $\Gamma(s)$  — слева (см. 9.2  
 случай Б).

Интеграл тогда вычисляется по теореме о Вычетах (для случая, когда контур  
 замкнут в правой полуплоскости):

$$I = -2\pi i \sum \text{Rez}(\Gamma(\dots)\dots) \quad (9.20)$$

Замыкать контур можно как слева, так и справа (но, необходимо учитывать знак  
 направления обхода контура).

Чему равен вычет для Гамма-функции? Из курса Квантовой теории поля:

$$\boxed{\Gamma(-n + \varepsilon) = \frac{(-1)^n}{n!} \left( \frac{1}{\varepsilon} + o(\varepsilon) \right)} \quad (9.21)$$

И так как вычет — это « $-1$ » член ряда Лорана, то  $\text{Rez}(\Gamma(-n + \varepsilon)) = \frac{(-1)^n}{n!}$ .

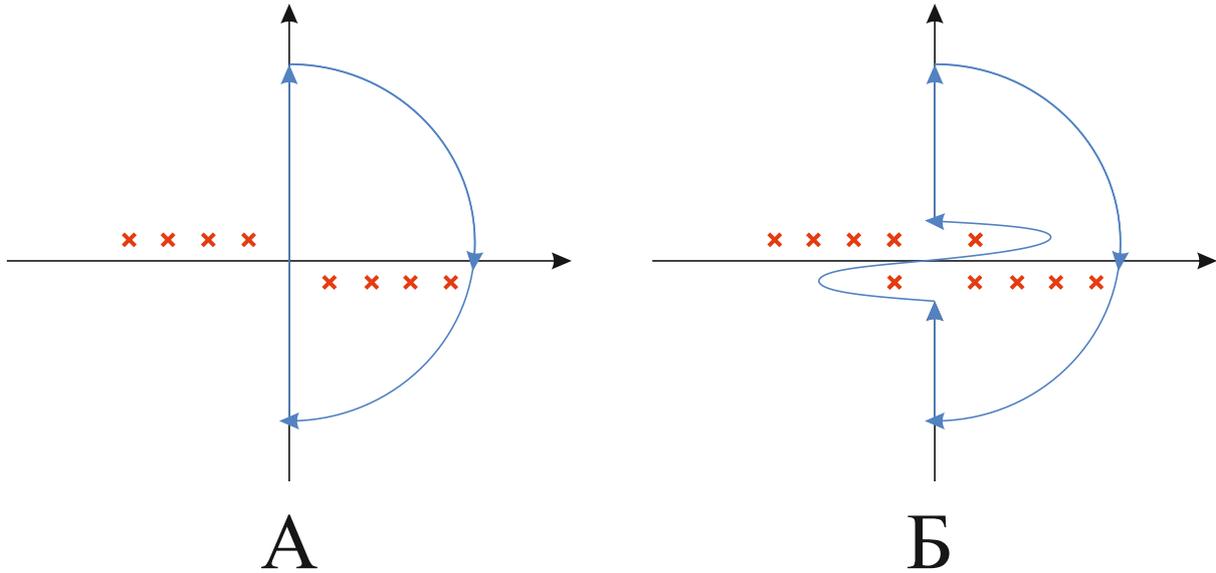


Рис. 9.2. Контуры обхода полюсов в различных случаях

## Применение метода Меллина-Барнса к вычислению интегралов Фейнмана

Рассмотрим интеграл в Фейнмановском представлении

$$I = \frac{e^{l\epsilon\gamma_E} \Gamma(\mathbf{v} - \frac{lD}{2})}{\prod_{j=1}^{n_{int}} \Gamma(\mathbf{v}_j)} \int_{\alpha_j > 0} d^{n_{int}} a \delta \left( 1 - \sum_{j=1}^{n_{int}} a_j \right) \prod_{j=1}^{n_{int}} a_j^{\mathbf{v}_j - 1} \frac{[U(a)]^{\mathbf{v} - \frac{l+1}{2}D}}{F(a)^{\mathbf{v} - lD/2}} \quad (9.22)$$

где  $\mathbf{v} = \sum_j \mathbf{v}_j$ . В случае, когда нет петель  $U = F = 1$ ,  $l = 0$ . Тогда интеграл  $I \equiv 1$ .  
 Можем формально написать

$$1 = \frac{\Gamma(\mathbf{v})}{\prod_j \Gamma(\mathbf{v}_j)} \underbrace{\int_{a_j > 0} d^{n_{int}} a \delta \left( 1 - \sum_{j=1}^{n_{int}} a_j \right) \prod_{j=1}^{n_{int}} a_j^{\mathbf{v}_j - 1}}_{\frac{\prod_j \Gamma(\mathbf{v}_j)}{\Gamma(\mathbf{v})}} \quad (9.23)$$

Интеграл должен быть равен  $\frac{\prod_j \Gamma(\mathbf{v}_j)}{\Gamma(\mathbf{v})}$ , чтобы удовлетворить тождеству.

Применяя преобразование Миллена-Барнса к полиному  $F(a)$

$$F(a) = \sum_{ij} t_{ij} a_i a_j = \sum_n^{N_{max}} \text{sym} \{t_{ij}\} a_i a_j \quad (9.24)$$

можем «перетащить» полином из знаменателя в числитель:

$$\begin{aligned} \frac{1}{(F(a))^{v-\frac{ID}{2}}} &= \frac{1}{\Gamma(v-\frac{ID}{2})} \frac{1}{(2\pi i)^{N_{\max}-1}} \times \\ &\times \int_{-i\infty}^{+i\infty} \prod_{i=1}^{N_{\max}-1} ds_i \Gamma(-s_i) \Gamma\left(s_1 - s_N + v - \frac{ID}{2}\right) \times \\ &\times \prod_{n=2}^{n_{\max}-1} \frac{((t_{ij})_n a_i a_j)}{\left((t_{ij})_{n_{\max}} a_i a_j\right)^{-\sum_i s_i - v + \frac{ID}{2}}} \end{aligned} \quad (9.25)$$

Тогда весь набор параметров  $F$  оказывается в числителе. Для них можно применить формулу (9.23) и расписать все в виде  $\Gamma$ -функций. Такое выражение можно вычислить численно.

Кроме того, при помощи т.н. лемм Барнса можно взять некоторые интегралы аналитически.

## Леммы Барнса

**Утверждение 1:**

$$\boxed{\frac{1}{2\pi i} \int_{-i\infty}^{+i\infty} dz \Gamma(a+z) \Gamma(b+z) \Gamma(c-z) \Gamma(d-z) = \frac{\Gamma(a+c) \Gamma(a+d) \Gamma(b+c) \Gamma(b+d)}{\Gamma(a+b+c+d)}} \quad (9.26)$$

**Утверждение 2:**

$$\boxed{\frac{1}{2\pi i} \int_{-i\infty}^{+i\infty} ds \frac{\Gamma(a+s) \Gamma(b+s) \Gamma(c+s) \Gamma(d-s) \Gamma(-s)}{\Gamma(e+s)} = \frac{\Gamma(a) \Gamma(b) \Gamma(c) \Gamma(d+a) \Gamma(d+b) \Gamma(d+c)}{\Gamma(e-a) \Gamma(e-b) \Gamma(e-c)}} \quad (9.27)$$

В процессе вычисления петлевого интеграла методом Меллина-Барнса складывается интеграл, схематичный вид которого

$$\iiint ds_1 \dots ds_n \frac{\Gamma \Gamma \Gamma \dots}{\Gamma \dots} x^s, \quad (9.28)$$

где  $\Gamma$ -функции могут зависеть от  $\frac{p^2}{m^2}$ ,  $\frac{p^2}{q^2}$ ,  $\varepsilon$  — размерность. Этот метод позволяет численно вычислить интеграл из разложения интеграла по  $\varepsilon$

$$I = \frac{(\dots)}{\varepsilon^n} + \frac{(\dots)}{\varepsilon^{n-1}} + \dots + \frac{(\dots)}{\varepsilon} + \dots + ((\dots)) \varepsilon^b. \quad (9.29)$$

## Связь Гамма-функций с гипергеометрическими функциями

Кроме того, нужно отметить, что Гамма-функция связана с гипергеометрическими функциями. Рассмотрим гипергеометрическую функцию Гаусса :

$$\begin{aligned} {}_2F_1(a, b, c, z) &= \frac{\Gamma(c)}{\Gamma(a)\Gamma(b)} \frac{1}{2\pi i} \int_{-i\infty}^{+i\infty} \frac{\Gamma(a+s)\Gamma(b+s)\Gamma(-s)}{\Gamma(c+s)} (-z)^s ds = \\ &= -\frac{\Gamma(c)}{\Gamma(a)\Gamma(b)} \sum_n \frac{\Gamma(a+n)\Gamma(b+n)}{\Gamma(c+n)} \frac{z^n}{n!} \end{aligned} \quad (9.30)$$

Переход последней строчке обусловлен теоремой вычетов, в случае, когда контур замыкается в правой полуплоскости. Полюса внутри контура  $s = 0, 1, 2, 3, \dots$

Вводя символ Похгаммера

$$(a)_n = (a+n)(a+n-1) = \frac{\Gamma(a+n)}{\Gamma(a)} \quad (9.31)$$

можем выразить гипергеометрическую функцию Гаусса в виде

$${}_2F_1(a, b, c|z) = \sum_n \frac{(a)_n (b)_n}{(c)_n} \frac{z^n}{n!} \quad (9.32)$$

Что и является определением гипергеометрической функции Гаусса через ряд.

Можно рассмотреть также интегральное представление гипергеометрической функции Гаусса:

$${}_2F_1(a, b, c|z) = \frac{\Gamma(c)}{\Gamma(b)\Gamma(c-b)} \int_0^{\infty} t^{b-1} (1-t)^{c-b-1} (1-tz)^{-a} dt \quad (9.33)$$

и при  $z = 1$

$${}_2F_1(a, b, c|1) = \frac{\Gamma(c)\Gamma(c-b-a)}{\Gamma(c-b)\Gamma(c-a)}. \quad (9.34)$$

Именно на этих приемах — сведению к Гамма-функциям и гипергеометрии, основано численное вычисление интегралов Фейнмана.

# Лекция 11. Численные методы вычисления интегралов Фейнмана. Метод разложения по секторам

Ранее мы рассматривали, как интегралы можно численно посчитать при помощи метода Меллина-Барнса. Существует несколько вариантов технической реализации этого метода — MB, AMBRE, MBResolve.

На этом занятии будет описан еще один метод численного взятия интегралов — *разложение по секторам* (*sector decomposition*). Соответствующие программы — `pysecdec`, `FIESTA`.

## Метод разложения по секторам

Как и всегда, будем рассматривать только те интегралы, которые регуляризуются размерной техникой. Для таких интегралов хорошо определено т.н. *представление Фейнмана* для интегралов:

$$I = \frac{e^{i\gamma\epsilon} \Gamma(v - \frac{lD}{2})}{\prod_{j=1}^n \Gamma(v_j)} \int_{v_j > 0} \prod_{j=1}^n da_j a_j^{v_j - 1} \delta\left(1 - \sum_{j=1}^n a_j\right) \frac{U(a)^{v - \frac{(l+1)D}{2}}}{F(a)^{v - \frac{lD}{2}}}, \quad (10.1)$$

где  $\epsilon$  — параметр размерной регуляризации ( $D = 4 - 2\epsilon$ ), где  $n$  — количество Фейнмановских параметров, равное количеству внутренних линий, а  $v_j$  — степени каждого из пропагаторов,  $v = \sum_{j=1}^n v_j$  — сумма всех степеней,  $U(a)$  и  $F(a)$  — полиномы Сигманзика.

Несмотря на то, что мы регуляризовали теорию, в интеграле (10.1) все еще могут оставаться сингулярности, связанные с тем, что степени  $v_j$  могут быть отрицательными. Именно для того, чтобы убрать эти «остаточные» сингулярности используется метод разложения по секторам.

Рассмотрим  $n$ -кратный интеграл, который можно разбить на  $n$ -секторов:

$$\int_{a > 0} d^n a = \sum_{k=1}^N \int_{a > 0} d^n a \prod_{\substack{l=1 \\ l \neq k}}^N \theta(a_k - a_l), \quad (10.2)$$

где  $\theta$ -функция Хевисайда:

$$\theta(x - y) = \begin{cases} 1, & x > y \\ 0, & x < y \end{cases} \quad (10.3)$$

В качестве примера можно рассмотреть двумерный случай:

$$\int_{x, y > 0} dx dy = \int_{x, y > 0} dx dy \theta(x - y) + \int_{x, y > 0} dx dy \theta(y - x) \quad (10.4)$$

Геометрическая интерпретация этой формулы представлена на рисунке (10.1) — область интегрирования разбивается на подобласти.

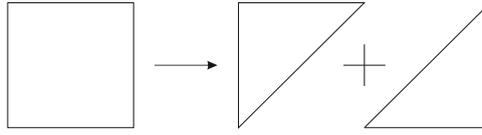


Рис. 10.1. Разбитие области интегрирования на подобласти

В случае 3-ех переменных, разбиение на сектора представляется в виде:

$$x, y, z \rightarrow \theta(x-y)\theta(x-z) + \theta(y-x)\theta(y-z) + \theta(z-x)\theta(z-y). \quad (10.5)$$

Применим подобное разбиение к интегралу Фейнмана, однако вводить новые  $\theta$ -функции в интеграл будет не очень удобно. Применим вспомогательный прием — введем новые переменные, которые будут связаны с исходными как:

$$a_j = \begin{cases} a_l t_j, & j > l \\ a_l, & j = l \\ a_l t_{j-1}, & j < l \end{cases} \quad (10.6)$$

где  $a_l$  — максимальная переменная в каждой подобласти, а  $t_j$  такие, что

$$0 < t_j < 1. \quad (10.7)$$

Интеграл в представлении Фейнмана перейдет в

$$I = \frac{e^{i\gamma_E \epsilon} \Gamma(v - \frac{D}{2})}{\prod_{j=1}^n \Gamma(v_j)} \sum_{l=1}^N I_l, \quad (10.8)$$

где интеграл в каждом секторе:

$$I_l = \int_0^1 \left( \prod_{j=1}^{N-1} dt_j t_j^{v_j-1} \right) da_l a_l^{N-1} a_l^{v-N} \delta \left( 1 - a_l \left( 1 + \sum_{j=1}^{N-1} t_j \right) \right) \times \quad (10.9)$$

$$\times \frac{a_l^{l(v - \frac{(l+1)D}{2})} U(t)^{v - \frac{(l+1)D}{2}}}{a_l^{(l+1)(v - D/2)} F(t)^{v - \frac{D}{2}}}, \quad (10.10)$$

и после сокращения всех коэффициентов  $a_l$  получим, что интеграл по  $da_l$  отфакторизовывается, и его можно вычислить отдельно

$$\int da_l \frac{1}{a_l} \delta \left( 1 - a_l \underbrace{\left( 1 + \sum_{j=1}^{N-1} t_j \right)}_{\equiv T} \right) = \int d(Ta_l) \frac{1}{Ta_l} \delta(1 - a_l T) = 1. \quad (10.11)$$

Теперь легко найти, чему равняется интеграл  $I_l$ :

$$I_l = \int_0^1 \left( \prod_{j=1}^{N-1} dt_j t_j^{v_j-1} \right) \frac{U(t)^{v-\frac{l-1}{2}D}}{F(t)^{v-\frac{lD}{2}}}. \quad (10.12)$$

Таким образом, мы избавились от зависимости от предыдущей переменной  $a_l$ . Теперь на каждой итерации, мы «выносим» из полиномов Симанзика один из коэффициентов, после чего, беря интеграл по  $\delta$ -функции, избавляемся от него. Продолжая действовать таким образом, мы можем свести все к ситуации, когда у нас не будет сингулярностей.

Выберем некоторый минимальный набор переменных  $(t_{a_1} \dots t_{a_r})$ , который бы был таким, чтобы при

$$t_{a_r} = 0 \implies U = 0, F = 0, \quad (10.13)$$

и для такого набора применим разделение на сектора:

$$\prod_{j=1}^r \theta(0 < t_{a_j} < 1) = \sum_{k=1}^r \prod_{\substack{j=1 \\ j \neq k}}^r \theta(t_{a_k} - t_{a_j}). \quad (10.14)$$

Аналогично можно ввести

$$t_{a_j} = \begin{cases} t_{a_k}, & k = j \\ t_{a_k} t_{a_j}, & k \neq j. \end{cases} \quad (10.15)$$

Действуя абсолютно аналогично, мы можем записать разложение, где  $I_l$ :

$$I_l = \sum_{k=1}^r G_{rl}, \quad (10.16)$$

где

$$G_{rl} = \int_0^1 \left( \prod_{j=1}^{N-1} dt_j t_j^{a_j+b_j\varepsilon} \right) \frac{U(t)^{v-\frac{(l+1)D}{2}}}{F(t)^{v-\frac{lD}{2}}}, \quad (10.17)$$

где коэффициенты  $a_j$  и  $b_j$  являются какой-то сложной комбинацией  $v, l, D$ . В силу однородности полиномов Симанзика, зависимость от параметра размерной регуляризации  $\varepsilon$  будет линейной.

Повторяем эту операцию, много раз до того момента, когда, при попытке найти минимальный набор  $(t_{a_1} \dots t_{a_r})$ , мы терпим неудачу. Тогда полиномы Симанзика представляют собой:

$$U = 1 + u(t); \quad (10.18)$$

$$F = s_0 + f(t), \quad (10.19)$$

$s_0$  — кинематический инвариант. Это является итогом применения метода разложения по секторам.

Таким образом, для каждого интеграла из первичного разбиения по секторам мы совершаем последовательное многократное разбиение по секторам, пока мы не найдем минимального набора  $t$  или, что идентично, выполнение условий (10.19). Тем самым, мы сводим выражение, содержащее сингулярности к регулярному.

Наша система разбиений представляет собой некий «ветвящийся граф». Общий интеграл является суммой выражений на последних ветках этого графа. Тогда можно ввести сплошную нумерацию через все эти выражения и полный интеграл

$$I = \sum_{l=1}^{N_{sec}} I_l, \quad (10.20)$$

$N_{sec}$  — общее число разбиений,  $I_j$  — финальные разложения интеграла в секторе.

*Замечание:* Подобные методы на математическом языке часто называют **blow up** — взрыв.

Рассмотрим внутри  $I_l$  интеграл по одной переменной  $t_j$ :

$$I_j = \int_0^1 dt_j t_j^{a_j+b_j\epsilon} \tilde{I}(t_j, \{t_{i \neq j}\}, \epsilon) \quad (10.21)$$

Рассмотрим 2 варианта:

- 1)  $a_j > -1$ . Тогда интеграл легко берется, и мы может перейти к следующей переменной  $t_{j+1}$ .
- 2)  $a_j < 1$ . Тогда регулярными методами взять интеграл не получится. Тогда необходимо разложить  $\tilde{I}$  в ряд Тейлора:

$$\tilde{I} = I(0, \{t_{i \neq j}\}, \epsilon) + \sum_{p=0}^{|a_j|-1} \frac{\tilde{I}^{(p)}(0, \{t_{i \neq j}\}, \epsilon)}{p!} t_j^p + R(t_j, t_i, \epsilon). \quad (10.22)$$

Подставляя такое разложение в интеграл  $I_j$  (10.21), получим

$$I_j = \sum_{p=0}^{|a_j|-1} \frac{1}{p!} \frac{\tilde{I}^{(p)}(0, \{t_i\}, \epsilon)}{a_j + b_j\epsilon + p - 1} + \int_0^1 dt_j R(t_j, \{t_i\}, \epsilon) t_j^{a_j+b_j\epsilon} \quad (10.23)$$

В качестве примера, можем рассмотреть случай,  $a_j = -1$ , когда ряд «обрезается» при  $p = 0$ , в такой ситуации

$$I_j = \frac{\tilde{I}}{b_j\epsilon} + \dots \equiv \sum_n \frac{a_n}{\epsilon^n} + \sum_n b_n + O(\epsilon). \quad (10.24)$$

Получаем непосредственное разложение по размерному параметру регуляризации.

Теперь, все сингулярности сводятся к сингулярностям по  $\epsilon$ , как и принято в Квантовой теории поля.

## Проблемы метода разложения по секторам

Основная проблема метода разложения по секторам заключается в неоднозначности выбора подвыборки  $S(t_1, \dots, t_r)$ . В зависимости от ее выбора может реализоваться бесконечная рекурсия.

**Пример:** реализации проблемы бесконечной рекурсии.

Рассмотрим набор из 3 параметров:  $(x_1, x_2, x_3)$  и полином вида:

$$f = x_1^2 + x_2^2 x_3. \quad (10.25)$$

Выберем  $S = \{1, 3\}$ , и рассмотрим сектор  $x_3 > x_1$ . Тогда  $x_1 = x_3 \cdot t_1$ . Получаем, что

$$f = x_3^2 t_1^2 + x_2^2 t_3. \quad (10.26)$$

Выберем  $S = \{2, 3\}$  и сектор  $x_3 > x_2$ . Тогда  $x_2 = x_3 \cdot t_2$ , и

$$f = x_3^2 t_1^2 + x_3^3 t_2^2 \rightarrow t_1^2 + x_3 t_2^2 \quad (10.27)$$

Сравнивая это выражение с исходным (10.25), можем понять, что они идентичны по своей структуре. Тогда, в результате действия этого алгоритма, мы всегда будем получать один и тот же полином в бесконечном цикле.

К этой проблеме есть различные подходы. Один из самых простых — располагая порядок в выборке  $S$  лексикографически, надо стараться, чтобы на следующей итерации выборка была меньше.

## Лекция 12. Программы AMBRE и FIESTA 4

Это занятие будет посвящено технической реализации методов, которые обсуждались на двух прошлых занятиях. Мы рассмотрим две программы — пакета для *Mathematica*, которые посвящены численному вычислению диаграмм Фейнмана.

### Пакеты AMBRE и MB

Эта программа позволяет производить преобразование Меллина-Барнса для интегралов Фейнмана автоматически.

- *Замечание:* Важно отметить, что эта программа использует не только *Mathematica*, но и другие системы компьютерной алгебры. Но подгружать эти дополнительные средства необязательно — программа может работать и в «чистой» *Mathematica*. Это будет замедлять выполнение программы, но не более чем.
- *Замечание:* Программа AMBRE совершает только преобразование Меллина-Барнса. Чтобы еще и вычислять интегралы Фейнмана, необходимо установить пакет MB.

Программа была написана *M.Czakon*, сейчас ей продолжают заниматься Владимир и Александр Смирновы.

### Установка пакета AMBRE и MB

- 1) Скачать пакет можно с *gitlab*'а проекта по ссылке <https://gitlab.com/feynmanintegrals/MB>.
- 2) Основные необходимые файлы — *MBresolve.m* и *MB.m*, которые можно скачать отдельно.
- 3) Кроме того с сайта проекта AMBRE <https://jgluza.us.edu.pl/ambre/>, надо скачать файл *AMBREv.2.1.1*, ссылка на который есть на главной странице.
- 4) Все эти файлы необходимо загрузить в соответствующую папку `AppData/Roaming/Mathematica/Applications`.

### Работа в программе AMBRE

- 1) Инициализируем пакет:

```
<< AMBREv2.1.1.m
```

2) Зададим инварианты

$$\text{invariants} = \{p_1^2 \rightarrow 0, p_2^2 \rightarrow 0, p_1 + p_2 \rightarrow s/20\};$$

3) Зададим размерность пространства (в рамках размерной регуляризации)

$$d = 4 - 2 \text{ eps};$$

4) Зададим выражение. Для этой программы выражение выглядит следующим образом:

$$\text{ex} = \text{PR}[k, m_1, n_1] \times \text{PR}[k, m_2, n_2]$$

оно соответствует произведению двух пропагаторов PR, каждый из которых включает в себя  $k$  — импульс,  $m$  — массу и  $n$  — степень выражения, которая пока не задана.

Такое выражение мы взяли, чтобы метод Меллина-Барнса правильно сработал.

5) Переходим в представление Меллина-Барнса для нашего выражения. В пакете AMBRE это соответствует команде

$$\text{repr} = \text{Mbrepr}[\{1\}, \{\text{ex}\}, \{k\}]$$

Полученный результат соответствует интегралу по параметрам  $z$ :

$$\int dz_1 (-1)^{n_1 + n_2} (m_1)^{z_1} m_2^4 (m_2^2)^{-\varepsilon - n_1 - n_2 + z_1} \times \frac{\Gamma(2 - \varepsilon - n_1 - z_1) \Gamma(-z_1) \Gamma(n_1 + z_1) \Gamma(-2 + \varepsilon + n_1 + n_2 + z_1)}{\Gamma(2 - \varepsilon) \Gamma(n_1) \Gamma(n_2)} \quad (11.1)$$

6) Чтобы перейти к численному интегрированию, необходимо задать параметры

$$\text{finres} = \text{repr} /. \{n_1 \rightarrow 1, n_2 \rightarrow 1\}$$

7) Запускаем пакеты MB:

```
<< MB.m  
<< MBresolve.m
```

- 8) Упрощаем полученное выражение, разлагаем его по степеням размерного параметра

```
step1 = MBresolve[#,eps] &/0 finres // Flatten  
step2 = MBexpand[step1, Exp[2 + eps + EulerGamma], {eps, 0, 0}]  
MBanalytic = MBmerge[step2]
```

- 9) Численное интегрирование выражения:

```
MBintegrate[MBanalytic, {m1 -> 1, m2 -> 1}, MaxNIntegrateDim -> 1]
```

- 10) Полученное выражение:

$$0.577216 + \frac{1}{\varepsilon} \quad (11.2)$$

Первое число соответствует  $\gamma_E$ .

## Программа FIESTA 4

### Установка пакета FIESTA 4

- 1) Необходимо скачать последнюю версию программы с сайта <https://gitlab.com/feynmanintegrals/fiesta>.
- 2) Основная программа — FIESTA5.m на момент записи видео.

### Работа в программе FIESTA 4

- 1) Открываем новый файл Wolfram Mathematica.
- 2) Инициализируем библиотеку командой

```
<<FIESTA4.m
```

- 3) Отключим дополнительные библиотеки командой, чтобы работать на «чистой» Mathematica.

UsingQLink = False; UsingC = False;

4) Задаем размерность пространства:  $d0 = 4$ ;

5) Основная команда в этом пакете SDEvaluate. Введем

SDEvaluate[UF[{k}, {-k^2+m},{m->1}], {1}, 0]

Обсудим отдельно параметры этой команды:

- UF — показывает, что полиномы Симанзика генерируются автоматически (при другом выборе, можно задавать полиномы самостоятельно);
- {k1, ...} — импульсы интегрирования в данном случае, он один;
- {-k^2+m, ...} — список пропагаторов;
- {m->1, ...} — поскольку интегрирование численное, мы должны задать значения параметров;
- {1, ...} — степени пропагаторов;
- 0 — задаем порядок разложения по  $\epsilon$ . Если этот параметр равен нулю, то разложение будет производиться до  $O(1)$ .

6) Полученный результат —  $-1. - 1/(\epsilon)$ .

Действительно, из квантовой теории поля, следует, что соответствующий интеграл равен:

$$J_1(p^2) = \frac{-i\lambda}{(2\pi)^{4-2\epsilon}} \frac{i}{2} \int \frac{d^{4-2\epsilon}k (\mu^2)^\epsilon}{k^2 - m^2}, \quad (11.3)$$

с точностью до численных множителей. Кроме того, команда SDEvaluate откидывает множители типа  $e^{-l\gamma_E}$ . С учетом этого ответ для этого интеграла:

$$J_1(p^2) = \frac{i\lambda}{32\pi^2} m^2 \left[ \frac{1}{\epsilon} + 1 - \gamma_E + \ln(4\pi) - \ln \frac{m^2}{\mu^2} \right] \quad (11.4)$$

соответствует тому, что посчитала программа.

*Замечание:* Для  $m \rightarrow 2$  ответ программы

$$-1.54518 - 4./\epsilon,$$

что в точности соответствует значению интеграла 11.4 при  $m = 2$ .

Аналогичным образом можно посчитать интеграл, когда пропагатор в знаменателе стоит во второй степени:

$\text{SDEvaluate}[\text{UF}[\{k\}, \{-k^2+m\}, \{m>1\}], \{2\}, 0]$

Тогда вывод программы

$-1./\epsilon p$

Интересно сравнить этот результат, с результатом полученным в программе **AMBRE**.

## Лекция 13. Особенности вычисления диаграмм в неабелевых калибровочных теориях

Тема сегодняшней лекции «стоит особняком», относительно других. Однако она существенна, и затрагивает вопрос о том, как работать с групповыми множителями, которые возникают в результате вычислений в неабелевых теориях.

### Диаграммы в неабелевых калибровочных теориях

Стандартной Модели соответствует калибровочная группа

$$G = U(1) \times SU(2) \times SU(3), \quad (12.1)$$

и, из-за наличия в ней подгруппы  $SU(3)$ , в которой генераторы сложны, необходимо знать, как выглядят правила Фейнмана с генераторами.

*Замечание:* Пакет для работы с генераторами группы  $SU(3)$  называется `color`.

Рассмотрим член в Лагранжиане, ответственный за взаимодействие для хромодинамики:

$$e\bar{\psi}\gamma^\mu A_\mu \psi \quad (12.2)$$

где  $A_\mu = ie(t)^a A_\mu^a$ ,  $a$  — групповой индекс. Возникает вопрос о виде генераторов  $t^a$ . Зачастую калибровочное поле лежит в присоединенном представлении алгебры Ли группы  $G$ . Тогда разложение по генераторам присоединенного представление обозначается как

$$A_\mu = ieT^A A_\mu^A. \quad (12.3)$$

Фермионные поля  $\bar{\psi}$  и  $\psi$ , как правило, лежат в фундаментальном представлении  $G$ . Тогда член взаимодействия представляется в виде свертки по групповым индексам как:

$$e\bar{\psi}_i \gamma^\mu A_\mu^A (T^A)^i_j \psi^j. \quad (12.4)$$

Значит, каждой вершине соответствует множитель, в котором будет генератор  $T^A$ .

Так, в случае самодействия в хромодинاميке — трехглюонной вершины, (см. рис. 12.1), вклад в диаграммы оказывается пропорциональным

$$\sim f^{ABC} \hat{D}_1^A A_\mu^A \hat{D}_2^B A_\mu^B \hat{D}_3^C A_\mu^C, \quad (12.5)$$

где  $\hat{D}_i$  — некие операторы, действующие на поля,  $f^{ABC}$  — структурные константы.

Таким образом, в общем случае, когда у нас есть диаграмма, то каждой вершине соответствует свой генератор. В итоге, вся конструкция будет пропорциональна произведению генераторов из каждой вершины.

$$\text{Диаграмма} \sim T^A T^B T^C \dots f f f \quad (12.6)$$

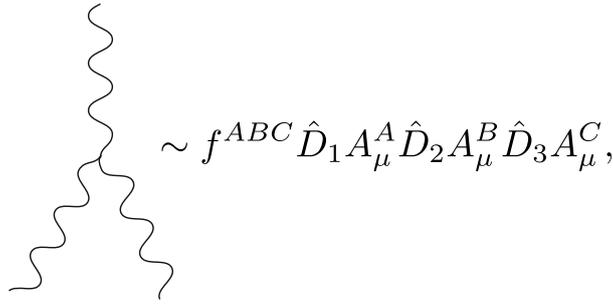


Рис. 12.1. Вершины в неабелевых калибровочных теориях

Часто бывает, что свертка генераторов образует след (например, в случае полной петли)  $\text{tr}(T^A T^B \dots)$ .

Задача состоит в том, чтобы преобразовать конструкции с генераторами к более удобному виду.

### Свойства генераторов калибровочной группы

Будем рассматривать 2 типа генераторов — лежащих в фундаментальном представлении  $(T_R^A)$  и в присоединенном представлении  $(T_{\text{adj}}^A)$ .

- 1) Самое главное свойство генераторов — их коммутация

$$\boxed{[T^A, T^B] = i f^{ABC} T^C}. \quad (12.7)$$

- 2) Генераторы присоединенного представления связаны со структурными константами как

$$(T_{\text{adj}})^A_{BC} = -i f^{ABC} \quad (12.8)$$

- 3) Введем величину

$$\text{tr} \left( T_R^{A_1} T_R^{A_2} \dots T_R^{A_n} \right) = M_R^{A_1 A_2 \dots A_n} \quad (12.9)$$

она формирует собой инвариантный тензор. Каждый генератор преобразуется под действием группы преобразований известным образом:

$$T_R^A \rightarrow U_R T_R^A U_R^{-1}, \quad (12.10)$$

где  $U_R$  — элемент группы. Если мы подставим эту конструкцию в уравнение (12.9), то можно убедиться, что под действием групповых преобразований след генераторов (а значит и тензор  $M_R$ ) остается инвариантным.

4) Введем объект, называемый оператором Казимира

$$C_R(M) = \sum_{A_1 \dots A_n} T_R^{A_1} \dots T_R^{A_n} M^{A_1 \dots A_n}, \quad (12.11)$$

где  $M^{ij\dots}$  — инвариантный тензор, определенный в предыдущем пункте.

**Пример:**

Самым простым инвариантным тензором является символ Кронекера  $\delta^{AB}$ :

$$\text{tr}(T_R^A T_R^B) \sim \delta^{AB}. \quad (12.12)$$

Отсюда можно сделать вывод, что один из операторов Казимира — это:

$$(C_R)_i^j = \left( T_R^A T_R^A \right)_i^j \quad (12.13)$$

В частном случае присоединенного представления:

$$C_{\text{adj}} \delta^{AB} = f^{ACD} f^{BCD}. \quad (12.14)$$

Стремимся к тому, чтобы записать все величины в задаче через операторы Казимира. На практике так удается сделать не всегда.

5) Введем величину, которая называется *симметризованный след*, которая по определению:

$$\text{Str} \left( T^{A_1} \dots T^{A_n} \right) = \frac{1}{n!} \sum_{\pi} \text{tr} \left( T^{A_1} \dots T^{A_n} \right), \quad (12.15)$$

где сумма берется по всем возможным перестановкам элементов под знаком следа. По определению вводится конструкция

$$S_R^{A_1 \dots A_n} = \text{Str} \left( T_R^{A_1} \dots T_R^{A_n} \right) = I_n(R) d^{A_1 \dots A_n} \quad (12.16)$$

Оказывается,  $S_R^{A_1 \dots A_n}$  представима в виде коэффициента  $I_n(R)$ , зависящего от числа генераторов и представления, и инвариантную относительно представления конструкцию  $d^{\dots}$ . Коэффициент  $I_n(R)$  называется индексом представления.

Так, в случае  $n = 2$  получим

$$\text{tr} \left( T^A T^B \right) = \text{Str} \left( T^A T^B \right) = I_2(R) \delta^{AB} \quad (12.17)$$

и в случае фундаментального представления,  $I_2 = \frac{1}{2}$ . Эти величины тоже часто появляются при вычислении диаграмм.

6) При вычислении следов от генераторов можно пользоваться формулой

$$\text{tr} \left( T^{A_1} \dots T^{A_n} \right) = \text{tr} \left( T^{A_1} \dots T^{A_n} \right) - \text{Str} \left( T^{A_1} \dots T^{A_n} \right) + d_R^{A_1 \dots A_n} \quad (12.18)$$

- Как минимум  $n$ -штук перестановок в первом и втором слагаемом в правой части формулы будут равны друг другу за счет цикличности следа.
- За конечное число шагов мы можем переставить местами генераторы, соответствующие различным индексам  $A_i$  и  $A_j$  (за счет свойств коммутации генераторов). Переставив генераторы, мы получим похожие выражения, плюс выражение  $\sim if^{ijk}T^k$ . Можно сказать, что таким образом мы «улучшили ситуацию» на 1 генератор. Таким образом, за конечное число шагов  $\text{tr}$  и  $\text{Str}$  сократят друг друга, и останутся только коммутаторы.

$$\text{tr} \left( T^{A_1} \dots T^{A_n} \right) \rightarrow \text{tr} \left( T^{A_1} \dots T^{A_{n-1}} \right) f + \text{tr}(\dots)ff + \dots + d_R^{A_1 \dots A_n} \quad (12.19)$$

- Повторяя эту процедуру, мы можем привести след от произвольного числа генераторов к следу от двух генераторов. Избавляясь от следов, мы «зарабатываем» структурные константы.

Проверим эту методику явно на примере следа 3-х генераторов:

$$\text{tr} \left( T^A T^B T^C \right) = \text{tr} \left( T^A T^B T^C \right) - \text{Str} \left( T^A T^B T^C \right) + d_R^{ABC} \quad (12.20)$$

Все возможные перестановки генераторов, где чертой обозначены циклические перестановки:

$$\overline{T^A T^B T^C}, T^A T^C T^B, T^B T^A T^C, \overline{T^B T^C T^A}, T^C T^B T^A, \overline{T^C T^A T^B}. \quad (12.21)$$

Для оставшихся, используя правила коммутации, получаем

$$T^A T^C T^B = T^C T^A T^B + if^{ACD} T^D T^B \quad (12.22)$$

$$T^B T^A T^C = T^A T^B T^C + if^{BAD} T^D T^C \quad (12.23)$$

$$T^C T^B T^A = T^C T^A T^B + if^{BAD} T^C T^D \quad (12.24)$$

Под симметричным следом все слагаемые идут с множителем  $\frac{1}{6}$ . Видно, что в (12.20) сократятся все следы трех генераторов. Останутся только следы от двух генераторов, которые уже известны. Можно убедиться, что

$$\text{tr} \left( T^A T^B T^C \right) = \frac{i}{6} I_2(R) \left( f^{ACB} + f^{BAC} + f^{BAC} \right) + d_R^{ABC} \quad (12.25)$$

Таким образом,

$$\boxed{\text{tr} \left( T^A T^B T^C \right) = d_R^{ABC} + \frac{i}{2} I_2(R) f^{BAC}}. \quad (12.26)$$

Этот алгоритм достаточно сложен в вычислительном плане. По этой причине, часто пользуются другим — более экономичным.

7) Существует формула

$$T_R^{\{A_1 \dots A_n\}} T_R^B = \sum_{j=0}^n \frac{(-1)^j}{j!} B_j E_j^{A_1 \dots A_n B}, \quad (12.27)$$

где  $B_j$  — числа Бернулли:

$$B_0 = 1, B_1 = -\frac{1}{2}, B_2 = \frac{1}{6}, \dots, \quad (12.28)$$

а  $E_j$ : такие, что

$$E_0^{A_1 \dots A_n} = T_R^{\{A_1 \dots T_R^{A_n}\}}, \quad (12.29)$$

где по индексам  $A_i$  производится симметризация,

$$E_j^{A_1 \dots A_n B} = \sum_{k=1}^n E_{j-1}^{A_1 \dots A_{i-1} A_{i+1} \dots A_n B} i^{A_i B} \quad (12.30)$$

Таким образом, рекурсивно сводим след от  $n$  генераторов, к следам от  $n-1$  генераторов.

Доказательство этой формулы сводится к доказательству выражения:

$$R_0 = \frac{1}{n+1} B_0 \sum_{i=0}^n R_i + \sum_{i=1}^n \frac{n!}{(n+1-i)!} \frac{(-1)^i}{i!} B_i \times \quad (12.31)$$

$$\times \sum_{j=0}^{i-1} (-1)^j \frac{(i-1)!}{j!(i-1-j)!} (R_j + (-1)^i R_{n-j}) \quad (12.32)$$

Эта формула справедлива для различных  $R_i$ . Будем понимать под  $R_0$  наше изначальное выражение

$$R_0 \equiv T_R^{\{A_1 \dots A_n\}} T_R^B, \quad (12.33)$$

под  $R_1$ :

$$R_1 \equiv T_R^{\{A_1 \dots T_R^B T_R^{A_n}\}}, \quad (12.34)$$

причем по индексу  $B$  симметризация не производится, и т.д.

Тогда, если учитывать правила коммутации генераторов, формула (12.33) сводится к изначальному утверждению (12.27).

Чтобы убедиться в правдивости этой формулы, рассмотрим некоторые частные случаи. Пусть  $R_0 \neq 0$ ,  $R_i \Big|_{i \neq 0} = 0$ , тогда,

$$R_0 = \frac{1}{n+1} B_0 \cdot R_0 + \sum_{i=1}^n \frac{n!}{(n+1-i)!} \frac{(-1)^i}{i!} B_i R_0. \quad (12.35)$$

Для  $B_n$  существует соотношение

$$B_n = -\frac{1}{n+1} \sum_{k=1}^n \binom{n+1}{k+1} B_{n-k} = -\frac{1}{n+1} \sum_{k=1}^{n-1} \binom{n+1}{k+1} B_{n-k} - \frac{1}{n+1} B_0 \quad (12.36)$$

где коэффициент:

$$\binom{n+1}{k+1} = \frac{(n+1)!}{(k+1)!(n-k)!} \quad (12.37)$$

С учетом этого, уравнение (12.35), обращается в верное тождество

$$R_0 = R_0. \quad (12.38)$$

Аналогично, можно убедиться, что формула обращается в верное тождество для

$$R_n \neq 0, R_0 = 0. \quad (12.39)$$

Наконец, для нашей изначальной конструкции:

$$\text{tr} \left( T_R^{\{A_1 \dots T_R^{A_n}\} T_R^B \right) = \sum_{j=0}^n \frac{(-1)^j}{j!} B_j E_j^{A_1 \dots A_n B} d_R^{A_1 \dots A_n B} \quad (12.40)$$

## Некоторые характерные конструкции, возникающие в неабелевых теориях

При рассмотрении реальных задач в неабелевых теориях, могут возникнуть конструкции:

1)

$$T_R^B T_R^A T_R^A T_R^C = C_R T_R^B T_R^C \quad (12.41)$$

2)

$$T^A T^B T^A = T^A \left[ T^B, T^A \right] + T^A T^A T^B = C(R) T^B + \underbrace{if^{BAD} T^A T^D}_{if^{BDA} T^D T^A} \quad (12.42)$$

Кроме того, можно расписать это выражение в форме

$$T^A T^B T^A = C(R) T^B + \frac{1}{2} if^{ABD} \underbrace{\left[ T^A, T^D \right]}_{=if^{ADE} T^E} = \left( C(R) - \frac{1}{2} C(A) \right) T^E \quad (12.43)$$

3) Наконец, в случае более сложной конструкции

$$T^A T^B T^C \dots T^A \quad (12.44)$$

можно протаскивать генератор  $T^A$  к началу, каждый раз «зарабатывая» структурную константу

$$T^A T^B T^C \dots T^A = C(R) T^B T^C + \dots + T^A T^B T^C \dots T^F i f^{AFE} \quad (12.45)$$

Последнее слагаемое также можно преобразовать, используя свойство из предыдущего пункта.

Такая ситуация, когда индексы в свертках генераторов повторяются, встречается в петлях, что очень сильно упрощает вычисления.

4) В итоге, после всех процедур, получится структура вида

$$\text{Диаграмма} \sim C(A)C(A)\dots ddd\dots fff. \quad (12.46)$$

В силу симметрии диаграмм, свертки структурных констант сформируют внутри себя несколько следов, что просто обработать системами компьютерной алгебры. Для этого необходимо свести все следы к простейшему случаю

$$\text{tr}(T_{\text{adj}}^A T_{\text{adj}}^D T_{\text{adj}}^I)^T = -\text{tr}(T_{\text{adj}}^I T_{\text{adj}}^D T_{\text{adj}}^A) = -\frac{1}{2} \text{tr}(T_{\text{adj}}^A [T_{\text{adj}}^D, T_{\text{adj}}^I]) = \quad (12.47)$$

$$= -\frac{1}{2} f^{DIA} C_{\text{adj}} \quad (12.48)$$

5) Аналогичным образом, след 5-ти генераторов:

$$F^{i_1 i_2 i_3 i_4 i_5} \equiv \text{tr}(T^{i_1} \dots T^{i_5}) \quad (12.49)$$

$$F^{i_1 i_2 i_3 i_4 i_5} = -F^{i_5 i_4 i_3 i_2 i_1} = -F^{i_1 i_5 i_4 i_3 i_2} = -F^{i_1 i_2 i_3 i_4 i_5} + F^{i_1 i_5 k} F^{k i_4 i_3 i_2} + \dots \quad (12.50)$$



ФИЗИЧЕСКИЙ  
ФАКУЛЬТЕТ  
МГУ ИМЕНИ  
М.В. ЛОМОНОСОВА

*teach-in*  
ЛЕКЦИИ УЧЕНЫХ МГУ