

МИНИСТЕРСТВО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ  
САМАРСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

Кафедра общей и теоретической физики

В.А. Салеев

**Система Mathematica**  
**для физиков**

*Учебное пособие для студентов физического факультета  
Рекомендовано НМС по физике УМО университетов России  
в качестве учебного пособия*

Издательство "Самарский университет"  
2000

УДК 681.3  
ББК 22.311  
С161

**Салеев В.А.** Система Mathematica для физиков: Учеб. пособие. Самара: Издательство "Самарский университет", 2000. 140 с.

ISBN 5-86465-141-9

В данном пособии можно найти описания основных функций системы Mathematica и прикладных пакетов Statistics, Graphics, Algebra, Calculus, Schroe и FeynCalc, применяемых для обработки результатов эксперимента, численного и аналитического решения алгебраических и дифференциальных уравнений, расчетов в квантовой механике и квантовой теории поля. Приведены примеры, которые демонстрируют возможности системы Mathematica для решения широкого круга физических задач. Предназначено для студентов и аспирантов физического факультета, преподавателей и сотрудников, использующих в своей работе систему Mathematica.

**Рецензент:** ведущий научный сотрудник НИИЯФ МГУ им.  
Д.В.Скобельцына д-р.физ.-мат. наук Н.П.Зотов

С  $\frac{1604010000 - 041}{6К4(03) - 2000}$  Без объяв.

ISBN 5-86465-141-9

©В.А. Салеев, 2000

©Изд-во "Самарский университет", 2000

# Содержание

Введение	6
<b>I. Основные возможности</b>	<b>8</b>
1. Оболочка системы и имена	8
2. Интерактивная помощь	9
3. Справочная база данных (Help)	10
4. Основные математические операции	10
5. Скобки и группирование выражений	10
6. Числовые множества	11
7. Элементарные численные выражения	12
8. Элементарные и специальные функции	16
9. Суммы и произведения	17
10. Алгебраические уравнения	19
11. Дифференциальные уравнения	21
12. Списки и массивы	22
13. Численные операции с данными	27
14. Графика	29
14.1. Графические опции	30
14.2. Построение графика по заданным точкам	32
14.3. Трехмерные графики	35
14.4. Параметрические графики	37
15. Векторы и матрицы	38

<b>16. Символьные вычисления</b>	<b>41</b>
16.1. Дифференцирование и интегрирование . . . . .	41
16.2. Ряды и пределы . . . . .	42
16.3. Полиномы и рациональные функции . . . . .	43
16.4. Операции с уравнениями . . . . .	45
<b>17. Программирование и замена переменных</b>	<b>47</b>
<b>18. Mathematica, T<sub>E</sub>X, Fortran, C</b>	<b>49</b>
<b>II. Дополнительные возможности</b>	<b>51</b>
<b>19. Пакет Statistics</b>	<b>51</b>
19.1. Операции со списками . . . . .	52
19.2. Обработка дискретного массива данных . . . . .	54
19.3. Непрерывные распределения . . . . .	56
19.4. Дискретные распределения . . . . .	56
19.5. Операции с распределениями . . . . .	57
<b>20. Пакет Algebra</b>	<b>59</b>
<b>21. Пакет Calculus</b>	<b>61</b>
21.1. $\delta$ -функция Дирака . . . . .	62
21.2. Преобразования Фурье . . . . .	63
21.3. Преобразования Лапласа . . . . .	66
21.4. Векторный анализ . . . . .	67
<b>22. Пакет Graphics</b>	<b>71</b>
22.1. Графика в различных системах координат . . . . .	71
22.2. Гистограммы . . . . .	72
22.3. Списки . . . . .	76
<b>23. Пакет FeynCalc</b>	<b>79</b>
23.1. Ввод функций . . . . .	79
23.1.1. Тензоры и скалярные произведения . . . . .	79
23.1.2. Матрицы Дирака . . . . .	81
23.1.3. Матрицы Гелл-Манна . . . . .	82
23.1.4. Спиноры . . . . .	82

23.1.5. Пропагаторы . . . . .	83
23.2. Элементарные вычисления . . . . .	84
23.2.1. Свертка по лоренцевским индексам . . . . .	84
23.2.2. Упрощение скалярных произведений 4-векторов	84
23.2.3. Упрощение произведений матриц Дирака и спиноров . . . . .	85
23.2.4. Вычисление следа произведения матриц Дирака	86
23.2.5. Вычисление следа матриц Гелл-Манна . . . . .	88
23.3. Отличия FeynCalc 3.0 от FeynCalc 1.0 . . . . .	89
<b>III. Примеры решения задач</b>	<b>92</b>
24. Обработка экспериментальных данных	92
25. Задачи по общей физике	103
25.1. Движение спутника Земли . . . . .	103
25.2. Электростатическое поле системы зарядов . . . . .	104
25.3. Основное уравнение электронной оптики . . . . .	108
26. Решение уравнения Шредингера	109
27. Вычисления в квантовой теории поля	116
27.1. Поляризационные эффекты в комптоновском рассеянии	116
27.2. Распад $\mu \rightarrow e\bar{\nu}_e\nu_\mu$ . . . . .	120
27.3. Процесс $gg \rightarrow Q\bar{Q}$ . . . . .	123
Предметный указатель	131

# Введение

В последние годы широкое развитие получили компьютерные системы численных математических вычислений, такие как **Eureka**, **MathLab**, **MatCAD** и др. Наряду с ними создан ряд систем компьютерной алгебры или символьной математики: **Reduce**, **SCONSHIP**, **MACSYMA** и др. Система **Mathematica**, а также **Maple** принадлежат к разряду универсальных компьютерных систем, в которых интегрированы возможности численных и аналитических вычислений, богатой компьютерной графики.

Недавно (конец 1999 года) фирма **Wolfram Research Incorporated** (<http://www.wri.com>) выпустила обновленную версию системы **Mathematica 4**. В настоящем пособии рассматривается базовая версия системы **Mathematica 2.2**, работающая как под управлением **Windows 95**, так и **Windows 3.11**. На уровне пользователя принципиальными отличиями **Mathematica 3.0** и **Mathematica 4** от предыдущей версии **Mathematica 2.2** являются следующие: более удобный графический интерфейс программы, включение в ядро системы ряда программ, которые ранее нужно было загружать дополнительно, новая справочная система **Help**.

В предлагаемом учебном пособии можно найти описание основных функций системы **Mathematica** и прикладных пакетов **Statistics**, **Graphics**, **Algebra**, **Calculus**, **Schroe** и **FeynCalc**, применяемых для обработки результатов эксперимента, численного и аналитического решения алгебраических и дифференциальных уравнений, расчетов в квантовой механике и квантовой теории поля. Приведены примеры, которые демонстрируют возможности системы **Mathematica** для решения широкого круга физических задач.

В качестве дополнительного справочного пособия на русском языке мы хотели бы рекомендовать книгу В.П. Дьяконова "Системы численной математики **Mathematica 2** и **Mathematica 3**" (М.: СК Пресс, 1998), которая по полноте материала не уступает фирменному руководству по системе **Mathematica 2.2** и комплекту электронной документации, распространяемому вместе с **Mathematica 4**.

Много полезной информации о системе **Mathematica** можно найти на сайте <http://www.wri.com>, включая описание большого числа прикладных пакетов, используемых в различных областях физики.

Как и всегда, освоение нового “инструмента” решения математических и физических задач требует дополнительных усилий и времени. Опыт автора показывает, что все это окупится с лихвой и достаточно быстро. **Mathematica** поможет вам забыть, что такое справочные пособия по интегралам, рядам и специальным функциям, лишит “удовольствия” написания собственных программ при решении алгебраических и обыкновенных дифференциальных уравнений, задач оптимизации, обработки массивов данных, поможет качественно и без хлопот представить любую графическую информацию. Обычно это происходит так: при решении какой-нибудь физической задачи вы написали уравнение или систему уравнений, если **Mathematica** не может решить эти уравнения аналитически, то это практически на 100% означает, что аналитического решения нет, запускайте процедуру численного решения, если вы некорректно сформулировали условия, то **Mathematica** даст об этом знать. Вас что-то не устраивает на этапе численных расчетов (чаще всего скорость вычислений) — сохраните аналитические результаты в формате **Fortran** или **C** и сделайте это быстрее (если сможете), чем **Mathematica**. Итак, от аналитических и численных расчетов до представления результатов в символьном или графическом виде, который допускает использование при оформлении дипломной работы или научной статьи различные графические форматы или формат популярного текстового редактора **TeX**— все это система **Mathematica**.

Часть разделов 19–22 и 24 были подготовлены О.Н. Пахомовой, которой автор особо благодарен за компьютерный набор и верстку представленного материала. Я также благодарен Р. Мертигу и Ф. Шоберлю за полезные консультации по пакетам **FeynCalc** и **Schroe**.

# I. Основные возможности

## 1. Оболочка системы и имена

Для выполнения простых арифметических операций в системе *Mathematica* достаточно набрать необходимое математическое выражение и нажать клавиши *Shift* и *Enter* одновременно (сама по себе клавиша *Enter* используется только для задания перевода строки) и программа выдает результат. Каждой входной/выходной строке или обращению присваивается свой номер. *N*-й запрос пользователя помечается *In[N]*, а *N*-й ответ начинается с *Out[N]*. Поэтому имеется возможность обращения к более ранним вычислениям. Для этого необходимо указать метку строки, к которой надо вернуться, например *In[3]*, и одновременно нажать *Shift* и *Enter*. Если после входной строки не стоит никаких знаков пунктуации, то на экране вы увидите выходную строку; в случае, когда после обращения пользователя стоит точка с запятой, выходная строка на экран не выводится, хотя подразумевается, что команда пользователя выполнена.

Каждая надпись, математическое выражение или график занимают отдельную ячейку - *Cell*. Это позволяет следить за тем, к чему относятся математические выражения - к исходным данным или результатам. Ячейка может занимать одну или несколько строк и всегда выделена своей квадратной скобкой (см. на правую часть главного окна редактирования). На ячейки налагаются ограничения: в каждой ячейке используется только один тип шрифта, нельзя комбинировать различные виды информации (например формулы и поясняющие надписи) и т.д. При малейшем изменении текста формул во входной ячейке эти изменения автоматически ведут к изменениям текстов последующих ячеек.

Имена всех функций, переменных, опций и констант, используемых в системе *Mathematica*, начинаются с заглавной буквы, например: *Integrate*, *Plot*. Если имя состоит из двух или большего количества слов, то первые символы каждого слова печатаются большими буквами: *ListPlot*, *NIntegrate*. Все имена объектов, сформированных в данной программе, — законченные слова, *Mathematica* редко использует сокращения. Обычно они появляются только там, где чрезвычайно общеизвестны.



Некоторые из сокращений приведены ниже:

- Abs* — абсолютное значение числа,
- Cos* — значение тригонометрической функции косинус,
- D* — вычисление производной,
- Det* — детерминант матрицы,
- GCD* — вычисление наибольшего общего делителя.

В соответствии с соглашением имени, определяемые пользователем, начинаются символом нижнего регистра (строчной буквой), например *myvar*, *var12* и т.д. Имена встроенных глобальных переменных обычно начинаются со знака \$, например, *\$Display*, *\$Context* и т.д. Не имеется никакого ограничения на длину имени.

## 2. Интерактивная помощь

*Mathematica* оперирует с более чем 800 встроенными функциями. Имя функции указывает цель выполняемой операции, например:

- Eigenvalues* — нахождение собственных значений,
- FindRoot* — численное решение уравнения,
- Integrate* — интегрирование,
- Timing* — показывает время, затрачиваемое для расчета выражения.

Если информации, заложенной в имени, не достаточно, можно воспользоваться встроенной интерактивной помощью. Интерактивная помощь используется для обращения к вспомогательной информации, включающей имена команд. Для этих целей служит специальный оператор, начинающийся со знака вопроса ?. Символ \*, используемый совместно с ?, обозначает групповой символ, т.е. любые алфавитно-цифровые последовательности. Если запросу соответствует более чем одна команда, то *Mathematica* выведет на экран все запрошенные команды. Например:

- ?var — показывает информацию об указанной переменной величине var,
- ??var — показывает дополнительную информацию,
- ?xyz\* — выдает список объектов, чьи имена начинаются с xyz,

- ?\*хуз\* — список всех объектов, содержащих в своем имени хуз,  
 ?@ — список объектов, чьи имена начинаются с символа нижнего регистра,  
 ?\* — выведет на экран все команды системы Mathematica.

### 3. Справочная база данных (Help)

Управление справочной базой данных осуществляется командами, входящими в меню *Help* (справка). Входящие в него команды и операции приведены в табл. 1.

Таблица 1

Команда или операция	Назначение команды подменю
<i>Contents</i>	Контекстно-зависимая справка
<i>Search for Help On</i>	Поиск по алфавиту
<i>How to use Help</i>	Правила работы со справочной системой
<i>About Mathematica</i>	Сообщение о системе и ее авторах
<i>Why the Beep</i>	Сообщение о работе со звуком

### 4. Основные математические операции

Приведем стандартные обозначения математических операций, поддерживаемых системой Mathematica:

- |   |                       |    |                   |
|---|-----------------------|----|-------------------|
| + | плюс,                 | !  | факториал,        |
|   | минус,                | >  | больше,           |
| * | умножение,            | >= | больше или равно, |
| / | деление,              | <  | меньше,           |
| ^ | возведение в степень, | <= | меньше или равно. |

### 5. Скобки и группирование выражений

Mathematica переняла различные формы условных обозначений, применяемые в языках высокого уровня, таких как C, FORTRAN, PASCAL. Примеры применения скобок в системе Mathematica можно свести в табл. 2.

Скобки	Назначение	Примеры
(term)	Группировка	$(a + b)/(c + d)$
f [expr]	Параметры функции	$\text{Sin}[2 \text{ Pi } x]$
{ a, b, c }	Определение вектора	{ x, 2 x, 3 x }
{{a1,b1},{a2,b2}}	Определение матрицы	{{1,2},{6,-3}}
v[i],w[i,j]	Введение индекса	m[3], m[1, 2]
(* remark *)	Комментарий	(*текст*)

## 6. Числовые множества

**Mathematica** может производить точные операции с целыми и рациональными числами. Вещественные числа рассматриваются в приближении, а комплексные числа могут быть как точные, так и приближенные в зависимости от задания действительной и мнимой частей. Табл. 3 содержит типы числовых множеств, поддерживаемых в системе:

Таблица 3

<i>Integer</i>	Целые	3
<i>Real</i>	Вещественные	3.4
<i>Rational</i>	Рациональные	3/4
<i>Complex</i>	Комплексные	$3 + 4.2i$
<i>Symbol</i>	Выражение или значение, зарезервированное за этим символом	<i>E</i>
<i>Infinity</i>	+ бесконечность	$+\infty$
<i>-Infinity</i>	- бесконечность	$-\infty$
<i>ComplexInfinity</i>	Число с бесконечной величиной, но с неопределенной мнимой частью	
<i>Indeterminate</i>	Выражение, величина которого не определена	

**Mathematica** распознает общеизвестные математические константы (Табл. 4).

Константа	Символ	Значение
$\sum_{k=0}^{\infty} (-1)^k / (2k + 1)^2$	<i>Catalan</i>	0.915966
$\pi/180$	<i>Degree</i>	0.0174453
<i>e</i>	<i>E</i>	2.71828
$\gamma$	<i>EulerGamma</i>	0.577216
$(1 + \sqrt{5})/2$	<i>GoldenRatio</i>	1.61803
<i>i</i>	<i>I</i>	$\sqrt{-1}$
$\pi$	<i>Pi</i>	3.14159

Символы *Catalan*, *Degree*, *E* имеют точные значения. *Mathematica* не станет вычислять их приблизительно, если вы не инструктируете ее делать это. Приблизительные значения для этих констант можно получать при использовании *N[...]*.

```
In[1] := N[Pi,10]
Out[1] = 3.141592654
```

## 7. Элементарные численные выражения

Численные вычисления *Mathematica* проводит подобно калькулятору. Выполняются арифметические операции типа: сложение, вычитание, умножение, деление и возведение в степень. Чтобы сложить два числа, необходимо набрать первое число, затем поставить знак + и второе число. Между числами и знаком плюс можно добавить пробелы для удобства чтения.

```
In[1] := 27+45
Out[1] = 72
```

Знак умножения \* необязателен. Пробел также обозначает умножение и выражение *a c* обозначает умножение *a* на *c*. Произведение чисел 127 и 9721 можно вычислять как 231 \* 5423, так и 231 5423.

В отличие от калькулятора при работе с целыми числами данная программа выдает точный результат, даже если он имеет большее количество цифр, чем может поместиться в одной строке:

```
In[2] := 21^43
Out[2] = 716852284415660510488171761482895729863
004676631934290061
```

Однако вычисление  $22/7$  даст

```
In[3] := 22/7
Out[3] = 22/7
```

Чтобы получить десятичную дробь, можно, например, поставить точку после 22 или 7, вычислив

```
In[4] := 22/7.
Out[4] = 3.14286
```

Другой способ — использование функции  $N[...]$ .

```
In[5] := N[22/7]
Out[5] = 3.14286
```

Команда  $N$  позволяет находить численное приближение для любых математических операций:

```
In[6] := N[Sqrt[17]]
Out[6] = 4.123056
```

Если требуется повышенная точность вычислений, то формат команды  $N[expr, m]$  предусматривает вычисление с точностью до  $m$  знаков. Например:

```
In[7] := N[Sqrt[17], 12]
Out[7] = 4.12310562562
```

Вещественное число можно представить в виде рациональной дроби командой  $Rationalize[x, dx]$ , где  $dx$  — необязательный параметр, задающий точность:

```
In[8] := Rationalize[9.6]
Out[8] = 48
--
5
```

```
In[9] := Rationalize[0.913043]
Out[9] = 0.913043
In[10] := Rationalize[0.913043, 0.000001]
Out[10] = 21
--
23
```

Часто используемые команды для операций с вещественными числами приведены в табл. 5.

Таблица 5

Команда	Результат выполнения
<i>Round</i> [ $x$ ]	Целое число, ближайшее к $x$
<i>Floor</i> [ $x$ ]	Целая часть числа $x$
<i>Ceiling</i> [ $x$ ]	Ближайшее целое число $> x$
<i>Sign</i> [ $x$ ]	1, если $x > 0$ и -1, если $x < 0$
<i>Abs</i> [ $x$ ]	Модуль числа $x$
<i>Max</i> [ $x_1, x_2, \dots, x_n$ ]	Максимально большое число из списка
<i>Min</i> [ $x_1, x_2, \dots, x_n$ ]	Минимальное число из списка

```
In[11] := Round[14.75]
Out[11] = 15
In[12] := Floor[14.75]
Out[12] = 14
In[13] := Ceiling[3.15]
Out[13] = 4
In[14] := Sign[-2]
Out[14] = -1
In[15] := Abs[-2]
Out[15] = 2
In[16] := Min[1,0,5,2,5,-3,4]
Out[16] = -3
In[17] := Max[1,0,5,2,5,-3,4]
Out[17] = 5
```

**Mathematica** работает с комплексными числами. Общая форма представления комплексного числа  $z$ :  $z = x + Iy$ , где  $x$  и  $y$  — вещественные числа. Некоторые команды для операций с комплексными числами приведены в табл. 6.

Таблица 6

Команда	Результат выполнения
<i>Re</i> [ $z$ ]	$x$ , вещественная часть числа $z$
<i>Im</i> [ $z$ ]	$y$ , мнимая часть числа $z$
<i>Conjugate</i> [ $z$ ]	$z^*$ , комплексно-сопряженное число
<i>Arg</i> [ $z$ ]	$\phi$ , фаза числа $z$
<i>Abs</i> [ $z$ ]	$ z $ , модуль числа $z$

```
In[18] := z=2.15+I 1.7
```

```
Out[18] = 2.15+1.7 I
```

```
In[19] := Re[z]
```

```
Out[19] = 2.15
```

```
In[20] := Im[z]
```

```
Out[20] = 1.7
```

```
In[21] := Conjugate[z]
```

```
Out[21] = 2.15-1.7 I
```

```
In[22] := Arg[z]
```

```
Out[22] = 0.669043
```

```
In[23] := Abs[z]
```

```
Out[23] = 2.74089
```

В системе существует встроенный генератор случайных чисел. Команда генерации случайных равномерно распределенных чисел может иметь различные форматы:

<i>Random</i> []	— генерирует вещественные случайные числа на отрезке $[0,1]$ ,
<i>Random</i> [ <i>Real</i> , $x_{max}$ ]	— генерирует вещественные случайные числа на отрезке $[0,x_{max}]$ ,
<i>Random</i> [ <i>Real</i> ,{ $x_{min},x_{max}$ }]	— генерирует вещественные случайные числа на отрезке $[x_{min},x_{max}]$ .
<i>Random</i> [ <i>Complex</i> ,{ $z_{min},z_{max}$ }]	— генерирует комплексные случайные числа, у которых $z_{min} <  z  < z_{max}$ ,
<i>Random</i> [ <i>Integer</i> ]	— генерирует случайные числа 0 и 1 с равной вероятностью,
<i>Random</i> [ <i>Integer</i> ,{ $I_{max},I_{min}$ }]	— генерирует целые случайные числа на отрезке $[I_{min},I_{max}]$ .

```
In[24] := Random[]
```

```
Out[24] = 0.165608
```

```
In[25] := Random[Real, -5, 0]
```

```
Out[25] = -4.88485
```

```
In[26] := Random[Complex, 2.15-I 1.7, 0]
```

```
Out[26] = 1.60614-0.810571 I
```

```
In[27] := Random[Integer, 2, 7]
```

```
Out[27] = 2
```

## 8. Элементарные и специальные функции

В системе *Mathematica* можно работать с большим количеством элементарных и специальных функций. За каждой из них закреплен определенный набор символов, который, как правило, соответствует общепринятому названию этой функции на английском языке (табл. 7).

Таблица 7

Функция	Стандартный вид	<i>Mathematica</i>
Экспонента	$e^x$	Exp[x]
Натуральный логарифм	$\ln(x)$	Log[x],
Логарифм по основанию b	$\log_b(x)$	Log[b,x]
Синус	$\sin(x)$	Sin[x]
n-факториал	$n!$	n!
"k" по модулю "n"	$\text{mod}_n(k)$	Mod[k,n]
Биномиальный коэффициент	$C_m^n$	Binomial[n,m]
Функция Бесселя	$J_n(z)$	BesselJ[n,z],
Полином Лагерра	$L_n^a(x)$	LaguerreL[n,a,x]

Численное значение элементарной или специальной функции при определенном значении аргумента может быть получено командой  $N[\text{Function}[\dots]]$ , например:

```
In[1] := N[Cos[Pi/8]]
Out[1] = 0.92388
In[2] := N[EulerE[4,0.5]]
Out[2] = 0.3125
```

Названия встроенных функций защищены и могут быть использованы пользователем в других целях только после снятия защиты командой  $\text{Unprotect}[\text{FunctionName}]$ . С помощью этой команды могут быть изменены свойства встроенных функций. Например, встроенная функция *Log* может быть дополнена правилом, по которому логарифм произведения представляется как сумма логарифмов:

```
In[3] := Log[x y z]
Out[3] = Log[x y z]
In[4] := Unprotect[Log];
        Log[x_,y_] := Log[x]+Log[y];
```



Protect[Log]

Out[4] = Log

In[5] := Log[a b c]

Out[5] = Log[a]+Log[b]+Log[c]

Кроме того, пользователь имеет возможность определить собственную функцию. Возможны два типа собственных функций: функция “присваивание”, которая задается командой

In[6] := userf[x\_]=Sin[x]\*Exp[-x]

Out[6] = Sin[x]\*Exp[-x]

In[7] := userf[y+z]

Out[7] = Sin[y+z]\*Exp[-x-z],

или функция “процедура”, которая задается командой

In[8] := userf[x\_] := .....

Например, вычисление  $n!$  можно организовать следующим образом:

In[9] := f[1]=1

Out[9] = 1

In[10] := f[n\_] := n f[n-1]

In[11] := f[10]

Out[11] = 3628800

## 9. Суммы и произведения

**Mathematica** позволяет вычислять конечные и бесконечные суммы и произведения элементов рядов, заданных аналитическими формулами.

- $Sum\{f, \{i, i_{max}\}\}$  — вычисляет сумму значений  $f$  при изменении индекса  $i$  от 1 до  $i_{max}$  с шагом +1,
- $Sum\{f, \{i, i_{max}, i_{min}\}\}$  — вычисляет сумму значений  $f$  при изменении индекса  $i$  от минимального значения  $i=i_{min}$  до максимального  $i=i_{max}$  с шагом +1,
- $Sum\{f, \{i, i_{max}, i_{min}, di\}\}$  — вычисляет сумму значений  $f$  при изменении индекса  $i$  от минимального значения  $i=i_{min}$  до максимального  $i=i_{max}$  с шагом  $di$ ,

$\text{Sum}[f, \{i, i_{\max}, i_{\min}\}, \{j, j_{\max}, j_{\min}, \dots\}]$  — вычисляет многократную сумму значений  $f$  при изменении индексов  $i$  от  $i=i_{\min}$  до  $i=i_{\max}$  с шагом  $+1$ ,  $j$  от  $j=j_{\min}$  до  $j=j_{\max}$  с шагом  $+1$  и т.д. (число индексных переменных неограничено).

In[1] := Sum[Cos[i^2-i], {i, 2, 5, 1}]

Out[1] = Cos[2] + Cos[6] + Cos[12] + Cos[20]

In[2] := Sum[1/i^2, {i, 1, Infinity}]  
-2

Out[2] = Sum[i, {i, 1, Infinity}]

$\text{NSum}[f, \{i, i_{\min}, i_{\max}\}]$  — возвращает численное значение суммы  $f[i]$  при индексе  $i$ , изменяющемся от  $i_{\min}$  до  $i_{\max}$  с шагом  $+1$ ,

$\text{NSum}[f, \{i, i_{\min}, i_{\max}, di\}]$  — возвращает сумму численных значений функции  $f[i]$  при индексе  $i$ , изменяющемся от  $i_{\min}$  до  $i_{\max}$  с шагом  $+1$ ,

$\text{NSum}[f, \{i, i_{\min}, i_{\max}\}, \{j, j_{\min}, j_{\max}, \dots\}]$  — выполняет многомерное суммирование.

Функция  $\text{NSum}[\dots]$  эквивалентна выражению  $N[\text{Sum}[\dots]]$ . Примеры:

In[3] := NSum[Cos[i^2-i], {i, 2, 5, 1}]

Out[3] = 1.79596

In[4] := NSum[1/i^2, {i, 1, Infinity}]

Out[4] = 1.64493

$\text{Product}[f, \{i, i_{\max}\}]$  — вычисляет произведение значений  $f[i]$  при изменении  $i$  от 1 до  $i_{\max}$  с шагом  $+1$ ,

$\text{Product}[f, \{i, i_{\min}, i_{\max}\}]$  — вычисляет произведение значений  $f[i]$  при изменении  $i$  от  $i_{\min}$  до  $i_{\max}$  с шагом  $+1$ ,

$\text{Product}[f, \{i, i_{\min}, i_{\max}, di\}]$  — вычисляет произведение значений  $f[i]$  при изменении  $i$  от  $i_{\min}$  до  $i_{\max}$  с шагом  $di$ ,

$\text{Product}[f, \{i, i_{\min}, i_{\max}, di\}, \{j, j_{\min}, j_{\max}, \dots\}]$  — вычисляет многократное произведение (по нескольким переменным).

In[5] := Product[Pi (i-1)/(i+1), {i, 2, 5, 0.5}]

7

Out[5] = 0.010101 Pi

Функция  $NProduct[...]$  эквивалентна выражению  $N[Product[...]]$ .

```
In[6] := NProduct[Pi (i-1)/(i+1),{i,2,5,0.5}]
Out[6] = 30.508
```

Приведенные функции **Mathematica** 2.2 не дают символического представления сумм и произведений, даже если оно существует. Для символического представления сумм и произведений можно использовать функции из дополнительного пакета **Algebra** (см. стр. 59).

## 10. Алгебраические уравнения

**Mathematica** включает в себя функции для нахождения корней уравнений в численном виде:  $NRoots$ ,  $NSolve$  и  $FindRoot$ . Символ  $==$  служит для обозначения равенства частей в уравнениях.

Функция  $NRoots[expr,x,n]$  находит приближенно корни полиномиального уравнения:

```
In[1] := NRoots[x^4-5 x^2 +x==3,x]
Out[1] = x== -2.43269, ||
        x== 0.0825028-0.732808I, ||
        x== 0.0825028+0.732808I, ||
        x== 2.26769, ||
```

Необязательный параметр  $n$  задает число цифр после запятой.

Здесь символ  $||$  — логическое **или**, т.е. указанную выше запись следует понимать так: если  $x$  равен  $-2.43269$  или  $x=0.08250-0.73281i$ , или  $x=0.088250+0.73281i$ , или  $x=2.26769$ , то уравнение  $x^4+3x^2+5x=7$  обращается в верное тождество.

Корни полинома можно также найти, используя функцию  $NSolve$ :

```
In[2] := NSolve[x^4-5x^2+x==3,x,10]
Out[2] = {x -> -2.43269},{x -> 0.0825028-0.732808 I},
        {x -> 0.0825028+0.732808 I},{x -> 2.267669}
```

В отличие от функции  $NRoots$ , функция  $NSolve$  может быть использована для численного нахождения корней произвольных алгебраических нетрансцендентных уравнений. Например:

```
In[3] := NSolve[Exp[-2*x]+5==0, x]
Out[3] = {{x -> -0.804719-0.157081I},
          {x -> -0.804719+0.157081I}}
```

Функция *NSolve* позволяет численно решать системы алгебраических уравнений:

```
In[4] := NSolve[{x^2+y-4==0, x+y^2-6==0},{x,y}]
Out[4] = {{x -> -2.63408, y -> -2.93838 },
          {x -> -1.15143, y -> 2.67422 },
          {x -> 1.35853, y -> 2.15441 },
          {x -> 2.42698, y -> -1.89024 }
```

Функция *FindRoot[lhs==rhs, {x,x0,x1,x2}]* ищет приближенное численное решение для уравнения  $lhs=rhs$ , начиная процедуру приближения с точки  $x=x_0$ , на интервале от  $x_1$  до  $x_2$  ( $x_1$  и  $x_2$  — необязательные параметры).

```
In[5] := FindRoot[Cos[x]/x==0, {x,2}]
Out[5] = { x-> 1.5708 }
In[6] := FindRoot[Cos[x]/x==0, {x,3,6,10}]
Out[6] = { x-> 7.85398 }
```

Видно, что если уравнение имеет много корней, то функция *FindRoot* находит лишь ближайший к точке  $x_0$  корень на заданном интервале.

В системе *Mathematica* существует возможность отыскания локального минимума явно заданной функции одной или нескольких переменных  $func\{x,y,\dots\}$  при помощи встроенной функции *FindMinimum[func, {x,x0},{y,y0},...]*, где  $x_0, y_0$  — координаты стартовой точки процедуры поиска минимума. Например:

```
In[7] := FindMinimum[Gamma[x],{x,2}]
Out[7] = { 0.8856603, x-> 1.46163 }
In[8] := FindMinimum[x^2+y^2+6x-7y,{x,2},{y,2}]
Out[8] = {-21.25, {x -> -3., y -> 3.5}}
```

Первая цифра в ответе - значение функции в точке локального минимума, затем приводятся координаты точки минимума.

*Mathematica* может интегрировать в численном виде, например, функция *NIntegrate[expr,{x,xmin,xmax}]* находит значение интеграла  $\int_{x_{min}}^{x_{max}} expr(x)dx$ . В *Mathematica 3.0* функция *NIntegrate* имеет полезную опцию *Method*, задающую метод интегрирования.

```

In[9] := NIntegrate[Sin[x]/x,{ x, 0, 1}]
Out[9] = 0.946083
In[10] := NIntegrate[Cos[x+y] Exp[-xy] ,{x,0,1},{y,x,1},
Method->MonteCarlo]
Out[10] = 0.222779

```

## 11. Дифференциальные уравнения

**Mathematica** численно решает системы дифференциальных уравнений. Используя команду `NDSolve`  $\{ \{ eqn_1, eqn_2, \dots \}, \{ y_1[x], y_2[x], \dots \}, \{ x, x_{min}, x_{max} \} \}$ , можем найти численное решение системы дифференциальных уравнений  $\{ eqn_1, eqn_2, \dots \}$  относительно функций  $y_1[x], y_2[x], \dots$  на отрезке  $[x_{min}, x_{max}]$ . 🍏

```

In[1] := NDSolve[{y''[x]-y'[x]+x^2 y[x]==0,
y'[0]==-1, y[0]==1},y,{x,0,5}]
Out[1] = {{ y-> InterpolatingFunction [ {0.,5.}, <>]}}

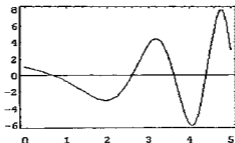
```

Команда `InterpolatingFunction[0.,5.,<>]` представляет интерполяционное решение исходного дифференциального уравнения на отрезке  $[0,5]$  с указанными начальными условиями. Можно изобразить это решение графически с помощью команды `Plot`:

```

In[2] := Plot[y[x]/.%,{x,0,5},{Frame->True}];

```



```

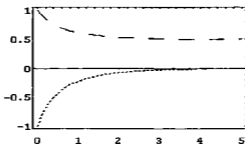
Out[2] = - Graphics -

```

Обозначение `y[x]/.%` заменяет функцию  $y[x]$  на выражение, полученное в предыдущем вычислении, а `Frame->True` предлагает установить рамку со значениями переменных  $x$  и  $y$  вдоль соответствующих осей. Подробное описание графических утилит пакета будет дано в разделе **Графика** (см. стр. 29).

Приведем пример численного решения системы двух уравнений с последующим представлением результатов в виде графика:

```
In[3] := equ1=y1'[x]+y2'[x]+y1[x] y2[x]==0
Out[3] = equ1=y1'[x]+y2'[x]+y1[x] y2[x]==0
In[4] := equ2=y1'[x]-y1[x] y2[x]==0
Out[4] = equ2=y1'[x]-y1[x] y2[x]==0
In[5] := R=NDSolve[{equ1,equ2,y1[0]==1,y2[0]==-1},
  {y1,y2},{x,0,5}]
Out[5] = {{y1->InterpolatingFunction[{{0.,5.},<>]}}
  {{y2->InterpolatingFunction[{{0.,5.},<>]}}
In[6] := Plot[{y1[x]/.R,y2[x]/.R},{x,0,5},Frame->True,
  PlotStyle->{Dashing[{0.07,0.07}],
  Dashing[{0.01,0.01}]}}
```



```
Out[6] = - Graphics -
```

## 12. Списки и массивы

Массив данных задается с помощью списка  $list = \{x_1, x_2, \dots, x_n\}$ , отдельный элемент списка  $list$  может быть получен с помощью команды  $list[[i]]$ .

```
In[1] := list={1,4,2,5,3,4,8}
Out[1] = {1,4,2,5,3,4,8}
In[2] := list[[3]]
Out[2] = 2
In[3] := list2={d,t,v,a,b,c}
Out[3] = {d,t,v,a,b,c}
In[4] := list2[[5]]
Out[4] = b
```

Система *Mathematica* имеет обширный набор функций для операций со списком. Приведем эти функции:

- Length[list]* — определяет число элементов списка *list*,  
*Drop[list,n]* — возвращает список *list*, из которого удалены первые *n* элементов,  
*Drop[list,-n]* — возвращает список *list*, из которого удалены последние *n* элементов.

```
In[5] := Length[list]
Out[5] = 7
In[6] := Drop[list,2]
Out[6] = {2, 5, 3, 4, 8}
In[7] := Drop[list,-3]
Out[7] = {1,4,2,5}
In[8] := Length[list2]
Out[8] = 6
In[9] := Drop[list2,3]
Out[9] = {a,b,c}
In[10] := Drop[list2,-1]
Out[10] = {d,t,v,a,b}
```

- Drop[list,{n}]* — возвращает список *list* без *n*-го элемента,  
*Drop[list,{m,n}]* — возвращает список *list*, отбросив элементы от *m* до *n*.

```
In[11] := Drop[list,{4}]
Out[11] = {1,4,2,3,4,8}
In[12] := Drop[list,{2,6}]
Out[12] = {1,8}
In[13] := Drop[list2,{2}]
Out[13] = {d,v,a,b,c}
In[14] := Drop[list2,{2,4}]
Out[14] = {d,b,c}
```

- Last[list]* — возвращает последний элемент списка *list*,  
*Rest[list]* — возвращает список с уничтоженным последним элементом.

In[15] := Last[list]

Out[15] = 8

In[16] := Rest[list]

Out[16] = {4,2,5,3,4,8}

In[17] := Last[list2]

Out[17] = c

In[18] := Rest[list2]

Out[18] = {d,t,v,a,b}

*Take*[list,n] — возвращает первые n элементов списка *list*,  
*Take*[list,-n] — возвращает последние n элементов списка *list*,  
*Take*[list,{m,n}] — возвращает элементы списка с порядковыми номерами от m до n.

In[19] := Take[list,4]

Out[19] = {1,4,2,5}

In[20] := Take[list,-2]

Out[20] = {4,8}

In[21] := Take[list,{2,5}]

Out[21] = {4,2,5,3}

In[22] := Take[list2,2]

Out[22] = {d,t}

In[23] := Take[list2,-3]

Out[23] = {a,b,c}

In[24] := Take[list2,{2,5}]

Out[24] = {t,v,a,b}

*Max*[ $x_1, x_2, \dots$ ] — возвращает наибольшее из  $x_i$ ,  
*Max*[{ $x_1, x_2, \dots$ }, { $y_1, y_2, \dots$ }, ...] — возвращает наибольший элемент любого из списков.

In[25] := Max[1,5,2,6.5,3,4]

Out[25] = 6.5

In[26] := Max[{1,2,3},{4,6,5},{9,8,7}]

Out[26] = 9

*Min*[ $x_1, x_2, \dots$ ] — возвращает наименьшее из  $x_i$ ,  
*Min*[{ $x_1, x_2, \dots$ }, { $y_1, y_2, \dots$ }, ...] — возвращает наименьший элемент любого из списков.



```

In[27] := Min[1,5,2,6.5,-3,4]
Out[27] = -3
In[28] := Min[{1,2,3},{4,6,5},{9,8,7}]
Out[28] = 1

```

Для генерации списков с элементами — вещественными и целыми числами или даже целыми выражениями — особенно часто используется функция *Table*, создающая таблицу-список:

<i>Table</i> [ <i>expr</i> ,{ <i>i</i> <sub>max</sub> }	— генерирует список, содержащий <i>i</i> <sub>min</sub> экземпляров выражения <i>expr</i> ,
<i>Table</i> [ <i>expr</i> ,{ <i>i</i> , <i>i</i> <sub>max</sub> }	— генерирует список значений <i>expr</i> при <i>i</i> , изменяющемся от 1 до <i>i</i> <sub>max</sub> с шагом 1,
<i>Table</i> [ <i>expr</i> ,{ <i>i</i> , <i>i</i> <sub>min</sub> , <i>i</i> <sub>max</sub> }	— генерирует список значений <i>expr</i> при <i>i</i> , изменяющемся от значения <i>i</i> = <i>i</i> <sub>min</sub> до значения <i>i</i> = <i>i</i> <sub>max</sub> с шагом 1,
<i>Table</i> [ <i>expr</i> ,{ <i>i</i> , <i>i</i> <sub>min</sub> , <i>i</i> <sub>max</sub> , <i>di</i> }	— генерирует список значений <i>expr</i> при <i>i</i> , изменяющемся от значения <i>i</i> = <i>i</i> <sub>min</sub> до значения <i>i</i> = <i>i</i> <sub>max</sub> с шагом <i>di</i> ,
<i>Table</i> [ <i>expr</i> ,{ <i>i</i> , <i>i</i> <sub>min</sub> , <i>i</i> <sub>max</sub> }, { <i>j</i> <sub>min</sub> , <i>j</i> <sub>max</sub> }	— генерирует двухмерный список (массив). Самым внешним является список по переменной <i>i</i> .

Примеры на использование функции *Table*:

```

In[29] := Table[2+3,{3}]
Out[29] = {5,5,5}
In[30] := Table[2+3*k,{k,5}]
Out[30] = {5,8,11,14,17}
In[31] := Table[2+3*k,{k,3,5}]
Out[31] = {11,14,17}
In[32] := Table[2+3*k,{k,3,5,0.5}]
Out[32] = {11,12.5,14.,15.5,17.}
In[33] := Table[2*m+k,{k,3,5},{m,0,2}]
Out[33] = {{3,5,7},{4,6,8},{5,7,9}}

```

Для расширения списка путем включения в него новых элементов используются следующие функции:

*Append*[list,element] — добавляет элемент в конец списка,  
*Prepend*[list,element] — добавляет элемент в начало списка,  
*Insert*[list,element,n] — вставляет элемент в позицию n (отсчет по-  
 зиции ведется с начала листа, а если задано -n,  
 то с конца), элемент может быть символом.

Приведенные ниже примеры иллюстрируют применение этих функций:

```
In[34] := Append[list,1]
Out[34] = {1,4,2,5,3,4,8,1}
In[35] := Prepend[list,2]
Out[35] = {2,1,4,2,5,3,4,8}
In[36] := Insert[list,e,5]
Out[36] = {1,4,2,5,e,3,4,8}
```

Иногда возникает необходимость комбинирования нескольких списков. Для этого используются:

*Join*[list1,list2,...] — объединяет списки в единую цепочку, может применяться на любом множестве выражений, имеющих один заголовок.

```
In[37] := list3={1,4,7}
          Join[list,list3]
Out[37] = {1,4,7}
Out[38] = {1,4,2,5,3,4,8,1,4,7}
```

Матрица задается как список списков по строкам:

```
In[39] := data={{1,2,6,3},{8,3,5,a},{8,5,3,4},{6,2,4,c}}
Out[39] = {{1,2,6,3},{8,3,5,a},{8,5,3,4},{6,2,4,c}}
```

Конечно, привычней ее видеть как матрицу:

```
In[40] := [MatrixForm[data]]
Out[40] = 1 2 6 3
           8 3 5 a
           8 5 3 4
           6 2 4 c
```

**Mathematica** позволяет создавать многомерные массивы — число элементов в них ограничено лишь объемом памяти компьютера.

Для задания массивов используют следующие функции:

- Array**[*f*,*n*] — генерирует список длиной *n* с элементами *f*[*i*], где *i* имеет значения от 1 до *n*,
- Array**[*f*,{*n*<sub>1</sub>,*n*<sub>2</sub>,...}] — генерирует массив с размерностью *n*<sub>1</sub>, *n*<sub>2</sub>,... в виде вложенных списков с элементами *f*[*i*<sub>1</sub>,*i*<sub>2</sub>,...],
- Array**[*f*,*dims*,*origin*] — генерирует список с размерностью *dims*, используя спецификацию индекса *origin*,
- Array**[*f*,*dims*,*origin*,*h*] — использует заголовок *h*, а не *List* для каждого уровня массива.

```
In[41] := Array[Exp, 4]
           2 3 4
```

```
Out[41] = {E, E , E , E }
```

```
In[42] := Array[f, {3, 4}]
```

```
Out[42] = {{f[1, 1], f[1, 2], f[1, 3], f[1, 4]},
           {f[2, 1], f[2, 2], f[2, 3], f[2, 4]},
           {f[3, 1], f[3, 2], f[3, 3], f[3, 4]}}
```

```
In[43] := Array[Sin, 3, 0]
```

```
Out[43] = {0, Sin[1], Sin[2]}
```

```
In[44] := Array[f, 5, 2, f10]
```

```
Out[44] = f10[f[2], f[3], f[4], f[5], f[6]]
```

### 13. Численные операции с данными

Экспериментальные данные представляются обычно в виде набора точек  $\{y_1, y_2, \dots\}$  (измеряемая величина рассматривается как функция номера измерения, т.е.  $x_1=1, x_2=2, \dots$ , набора пар точек  $\{(x_1, y_1), (x_2, y_2), \dots\}$  (измеряемая величина имеет значение  $y_i$  при  $x=x_i$ ) или набора трех точек  $\{(x_1, y_1, dy_1), (x_2, y_2, dy_2), \dots\}$ , где  $dy_i$  - погрешность измерения  $y_i$ . В системе **Mathematica** реализована процедура аппроксимации экспериментальных данных полиномом методом наименьших квадратов. Например:

*Fit*[[ $y_1, y_2, \dots$ ], {1,  $x$ },  $x$ ] — линейный фит,  
*Fit*[[ $y_1, y_2, \dots$ ], *Table*[ $x^j$ , { $j, 0, n$ }],  $x$ ] — аппроксимация полиномом  $n$ -й степени,  
*Exp*[*Fit*[*Log*[[ $y_1, y_2, \dots$ ]], {1,  $x$ },  $x$ ]] — аппроксимация данных зависимостью  $\exp(a+bx)$ .

```

In[1] := data=Table[Prime[k], {k, 10}]
Out[1] = {2, 3, 5, 7, 11, 13, 17, 19, 23, 29}
In[2] := f=Fit[data, {1, k}, k]
Out[2] = -3.26667+2.93939 k
  
```

В рассмотренном примере данные представляют собой набор первых 10 простых чисел, генерируемых командой *Prime*[ $k$ ], которые аппроксимируются полиномом первой степени ( $x=k$ ).

Для решения задач интерполяции и аппроксимации данных функций, заданных рядом узловых точек, используются следующие функции:

*InterpolatingPolynomial* [*data*, *var*] — определяет полином (степенной многочлен) по переменной *var*, дающий точные значения для данных списка *data*,  
*InterpolatingFunction* [*range*, *table*] — определяет аппроксимирующую функцию, чьи значения находятся при помощи интерполяции,  
*Interpolation*[*data*] — определяет объект *InterpolatingFunction*, который представляет собой приближенную функцию, интерполирующую данные *data*,  
*InterpolationOrder* — опция к функции *Interpolation*, которая указывает степень подходящего полинома.

```

In[3] := data=Table[Prime[k], {k, 10}]
         f1=Fit[data, {1, x}, x]
         f2=Interpolation[data]
         f3=InterpolatingPolynomial[data, x]
         g1=ListPlot[data, PlotStyle->PointSize[0.05]];
         g2=Plot[{f1, f2[x], f3}, {x, 0, 10}, PlotStyle->
         {Dashing[{0.07, 0.03}], Dashing[{0.015, 0.015}],
  
```

```
Dashing[{0.001,0.001}]]];
```

```
Show[g1,g2]
```

```
Out[3] = {2,3,5,7,11,13,17,19,23,29}
```

```
Out[4] = -3.26667+2.93939 x
```

```
Out[5] = InterpolatingFunction[{{1,10},<>]
```

```
Out[6] = 2 + (1 + (- + (-(-) + (- + (-(-) +
```

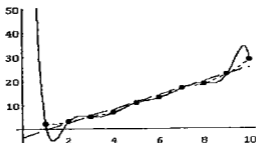
$$\frac{1}{2} + \frac{1}{6} + \frac{1}{8} + \frac{3}{40}$$

```
23 53 23 79 (-9 + x)
```

$$\left(\frac{---}{720} + \left(-\frac{---}{5040}\right) + \left(\frac{---}{8064} - \frac{---}{120960}\right) (-8 + x)\right)$$

```
(-7 + x)) (-6 + x)) (-5 + x)) (-4 + x)) (-3 + x))
```

```
(-2 + x)) (-1 + x))
```



```
Out[7] = - Graphics -
```

## 14. Графика

Графические возможности системы **Mathematica** способствуют ее широкому успеху. Используя данную программу, можно строить графики функций и выводить данные в двух- и трехмерном виде с помощью процедур *ListPlot*, *ListPlot3D*, *Plot3D*, *Show*, *Graphics*, *ParametricPlot*, *ContourPlot*.

В простейшем виде команда  $Plot[expr, \{x, x_{min}, x_{max}\}]$  строит график функции одной переменной. Трехмерные графики функций строятся при помощи команды

$$Plot3D[expr, \{x, x_{min}, x_{max}\}, \{y, y_{min}, y_{max}\}, Options].$$

## 14.1. Графические опции

При построении графиков в системе *Mathematica* предусмотрено большое количество изменяющихся параметров, таких как оттенки и цвет изображения, выбор осей и интервалов значений и т.д. Все эти установки меняются командой *Options*, справку по которой можно получить, набрав, как обычно, *?? Options*. Опции внутри записей графических функций задаются своим именем *Name* и значением *Value* в виде:

*Name* — *> Value*.

Значениями опций могут быть числа, списки, логические утверждения *True*, *False* и специальные слова, например:

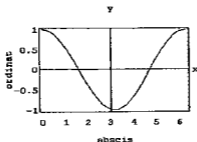
*Automatic* — *>* используется автоматический выбор,  
*None* — *>* опция не используется,  
*All* — *>* используется в любом случае.

Ниже представлены основные опции для графических функций (звездочкой отмечены те опции, которые можно применять и для трехмерной графики):

*AspectRatio\** — *>* задает пропорцию размеров графика — отношение высоты к ширине (по умолчанию задано *1/GoldenRatio*),  
*Axes\** — *>* устанавливает, должны ли рисоваться оси (по умолчанию *Automatic*),  
*AxesLabel\** — *>* устанавливает, нужны ли отметки на осях графиков (по умолчанию *None*),  
*AxesOrigin* — *>* указывает, где должно располагаться начало отсчета осей (по умолчанию *Automatic*),  
*Background\** — *>* указывает, какого цвета должен быть фон графика (по умолчанию *Automatic*),  
*ColorOutput\** — *>* указывает, каким цветом строится график (по умолчанию *Automatic*),  
*DefaultFont\** — *>* задает шрифт для надписей на графиках (по умолчанию *\$DefaultFont*),  
*Frame* — *>* указывает, нужно ли рисовать рамку вокруг графика (по умолчанию задано *False*),

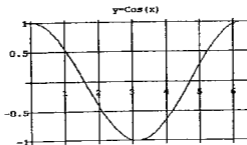
- FrameLabel* - > указывает, нужно ли на границах рамки, обрамляющей график, размещать метки (по умолчанию *None*),
- FrameStyle* - > указывает, в каком виде будут воспроизводиться линии рамки,
- FrameTicks* - > указывает плотность установки штриховых меток для границ рамки (по умолчанию *Automatic*),
- GridLines* - > указывает, нужно ли строить линии сетки графика (по умолчанию *None*),
- PlotLabel\** - > указывает, нужно ли вывести титульную надпись для графика (по умолчанию *None*),
- Ticks\** - > устанавливает штриховые метки для осей (по умолчанию *Automatic*)
- RGBColor* - > цвет.

```
In[1] := Plot[Cos[x],{x,0,2 Pi},AxesLabel->{"x","y"},
  AxesOrigin->{3,0},Frame->True,
  FrameLabel->{"abscis","ordinat"}]
```



Out[1] = - Graphics -

```
In[2] := g2=Plot[Cos[x],{x,0,2 Pi},AxesOrigin->{0,0},
  GridLines->Automatic,PlotLabel->"y=Cos(x)"]
```



Out[2] = - Graphics -

В неясных случаях для указания значений опций рекомендуется обращаться к функции *Options[Name]* или к оперативной справочной системе *Mathematica*.

Система использует директивы двумерной графики, которые указывают, с какими графическими параметрами (цвет, толщина линий, их стиль и т.д.) должна строить графики та или иная функция.

*AbsolutePointSize[d]* — устанавливает построение точек графика в виде кругов с абсолютным радиусом  $d$  (в пикселах),

*AbsoluteThickness[d]* — устанавливает толщину  $d$  рисуемых линий (в пикселах),

*Dashing[{r<sub>1</sub>, r<sub>2</sub>, ...}]* — устанавливает вывод линий пунктиром с последовательными сегментами длиной  $r_1, r_2, \dots$ , повторяющимися циклически.

## 14.2. Построение графика по заданным точкам

Строить графики часто приходится по точкам. Для этого существует встроенная в ядро графическая функция *ListPlot*:

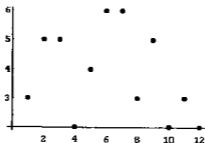
*ListPlot[{y<sub>1</sub>, y<sub>2</sub>, ...}]* — выводит график списка величин  $y_1, y_2, \dots$  и т.д. (координаты  $x$  для каждой точки принимают значения  $y_1, y_2, \dots$  и т.д.),

*ListPlot[{{x<sub>1</sub>, x<sub>2</sub>} {y<sub>1</sub>, y<sub>2</sub>}, ...}]* — выводит график списка величин с указанными  $x_i$ - и  $y_i$ -координатами ( $i=1, 2, \dots$  и т.д.).

*PointSize[r]* — указывает, что точки, нанесенные на график, должны изображаться в виде кругов с радиусом  $r$  (дробная часть от общей ширины графика).

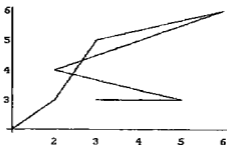
```
In[3] := ListPlot[{3, 5, 3, 2, 4, 6, 6, 3, 5, 2, 3, 2},  
PlotStyle->PointSize[0.05]]
```





Out[3] = - Graphics -

```
In[4] := g3=ListPlot[{{3,3},{5,3},{2,4},{6,6},{3,5},
  {2,3},{1,2}},PlotJoined->True]
```



Out[4] = - Graphics -

Отметим опцию для данной функции:

*PlotJoined* — указывает, следует ли точки, нанесенные на график, соединять отрезками прямых (по умолчанию — *False*).

При построении графиков приходится изменять их вид, а также те или иные параметры и опции. Для удобства используют специальные функции перестройки и вывода графиков, учитывающие, что точки этих графиков уже были рассчитаны и их координаты хранятся в памяти ПК.

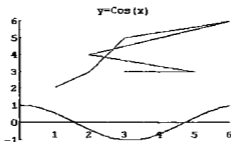
В этом случае удобно использовать следующие функции:

*Show[plot]* — построение графика по вычисленным данным,

*Show[plot, Option -> Value]* — построение графика с заданной опцией,

`Show[plot1,plot2]` — построение нескольких графиков с наложением их друг на друга.

```
In[5] := Show[g2,g3,GridLines->None,AxesOrigin->{0,0},  
PlotRange->{{0,6},{-1,6}}]
```



```
Out[5] = - Graphics -
```

Другое полезное применение этой функции — объединение воедино нескольких графиков различных функций или экспериментальных точек и графика теоретической зависимости.

Команда `Show[%, %%]` позволяет вывести одновременно два последних графика программы. Кроме того, ссылка в виде `%` может осуществляться и в любых других вычислениях системы **Mathematica**, как ссылка на последний результат.

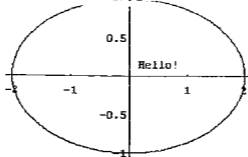
Текст на графике можно задать с помощью функции `Graphics[Text[expr, coords]]`, где `expr` — выражение в текстовой форме с центром, имеющим координаты `coords`. Например: `Graphics[Text["Hello!",0.5,0.15]]`.

Для построения параметрически заданных функций используют следующие графические функции системы **Mathematica**:

`ParametricPlot[{fx,fy}, {t,tmin,tmax}]` — строит параметрический график с координатами  $f_x$  и  $f_y$ , получаемыми как функции от  $t$ ,

`ParametricPlot[{{fx,fy}, {gx,gy},..., {t,tmin,tmax}]` строит графики нескольких параметрических кривых.

```
In[6] := g3=ParametricPlot[{x=2*Cos[t],y=Sin[t]},{t,0,2*Pi};  
g4=Graphics[Text["Hello!",{0.5,0.15}];  
Show[g3,g4]
```



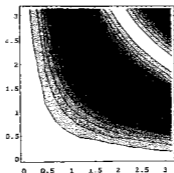
Out[6] = - Graphics -

### 14.3. Трехмерные графики

Команда *ContourPlot* строит проекцию трехмерного графика на плоскость  $XOY$ , выделяя одним цветом пересечение с плоскостями, параллельными координатной  $XOY$ , т.е. одинаковые значения функции.

*ContourPlot* $[f, \{x, x_{min}, x_{max}\}, \{y, y_{min}, y_{max}\}]$  — строит контурный график поверхности  $f(x, y)$ .

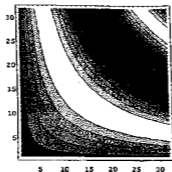
In[1] := *ContourPlot*[*Cos*[ $x+y$ ], { $x, 0, \text{Pi}$ }, { $y, 0, \text{Pi}$ }]



Out[1] = - Graphics -

*ListContourPlot* $[array]$  — строит контурный график для массива высот некоторой поверхности.

```
In[2] := data=Table[Sin[x y],{x,0,Pi,0.1},{y,0,Pi,0.1}];  
ListContourPlot[data]
```

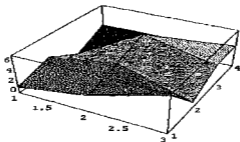


```
Out[2] = - Graphics -
```

Для построения графиков трехмерных поверхностей используются следующие основные графические функции:

- ListPlot3D[array]* — строит трехмерный график поверхности, представленной массивом *array*,
- ListPlot3D[array,shades]* — строит 3D-график так, что каждый элемент поверхности штрихуется согласно специфике *shades*.

```
In[3] := ListPlot3D[{{1,3,7},{3,5,1},{0,7,4},{3,3,4}}]
```

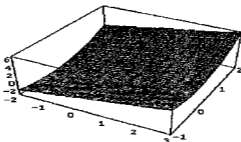


```
Out[3] = - Graphics -
```

$Plot3D[f, \{x, x_{min}, x_{max}\}, \{y, y_{min}, y_{max}\}]$ , — трехмерный график для аналитически заданной функции  $f(x, y)$  переменных  $x$  и  $y$ , изменяющихся в заданных пределах.

$Plot3D[\{f, s\}, \{x, x_{min}, x_{max}\}, \{y, y_{min}, y_{max}\}]$  — строит трехмерный график, в котором высоту поверхности задает аналитическая функция  $f$ , а затенение — параметр  $s$  (кроме того, задаются пределы изменения значений переменной  $x$  от  $x_{min}$  до  $x_{max}$  и переменной  $y$  от  $y_{min}$  до  $y_{max}$ ).

In[4] := Plot3D[x+y^2, {x, -2, 3}, {y, -1, 2}]



Out[4] = - Graphics -

$GraphicsArray[\{g_1, g_2, \dots\}]$  — представляет массив графических объектов,

$GraphicsArray[\{\{g_{11}, g_{12}, \dots\}, \dots\}]$  — представляет двумерный массив графических объектов.

## 14.4. Параметрические графики

Для построения 3D-графиков в параметрической форме также имеется ряд функций:

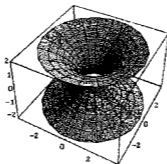
$ParametricPlot3D[\{f_x, f_y, f_z\}, \{t, t_{min}, t_{max}\}, \{u, u_{min}, u_{max}\}]$ , — строит трехмерную поверхность, параметризованную по  $t$  и  $u$  в заданных пределах их изменения,

$ParametricPlot3D[\{f_x, f_y, f_z\}, \{t, t_{min}, t_{max}\}]$  — выполняет построение трехмерной кривой, параметризованной по переменной  $t$ , которая изменяется от  $t_{min}$  до  $t_{max}$ ,

`ParametricPlot3D`[{ $f_x, f_y, f_z, s$ }, ...] — выполняет построение трехмерной поверхности, закрашенной в соответствии с цветовой спецификацией  $s$ ,

`ParametricPlot3D`[{ $f_x, f_y, f_z$ }, { $g_x, g_y, g_z$ }, ...], ...] — строит несколько 3D-поверхностей на одном графике.

```
In[1] := ParametricPlot3D[{Cosh[z] Cos[phi], Cosh[z] Sin[phi], z}, {z, -2, 2}, {phi, 0, 2 Pi}]
```



```
Out[1] = - Graphics -
```

## 15. Векторы и матрицы

`Mathematica` может работать с векторами и матрицами различной размерности. Вектор представлен как набор объектов, заключенных в фигурные скобки посредством следующей команды:

$$\{ item_1, item_2, item_3, \dots, item_n \}.$$

Величины  $item$  не обязательно принадлежат к одному и тому же типу. Следующий вектор содержит целое число, вещественное число и некоторые выражения:

$$\{ 4, 6.7, A, 7x+8, Sin[x] \}.$$

В системе **Mathematica** матрица задается как набор векторов. Например, матрица

$$\begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{pmatrix}$$

представлена как  $\{ \{ 1,2,3 \}, \{ 4,5,6 \} \}$ .

**Mathematica** содержит разнообразные встроенные функции, позволяющие оперировать с векторами и матрицами. Скалярное произведение и сумма двух векторов (матриц) находятся с помощью команд `+` и `.` соответственно.

```
In[1] := vecA={2x,y,5z}
In[2] := vecB={-x,2y,-3z}
In[3] := vecA.vecB
          2 2 2
Out[3] = -2x +2y -15z
```

Следует отметить, что команда умножение `*` в результате выдает вектор, в котором каждая координата результирующего вектора равна произведению соответствующих координат сомножителей, т.е. пользоваться этой командой для векторного произведения в трехмерии нельзя.

Для заданной матрицы можно найти матрицу, обратную ей:

```
In[4] := m={{1,2,3},{10,5,45},{17,6,7}}
In[5] := Inverse[m]
Out[5] = {{-47/216,1/270,5/72},{139/216,-11/270,-1/72},
          {-5/216,7/270,-1/72}}
```

а также собственные вектора и собственные значения:

```
In[6] := Eigenvalues[N[m]]
Out[6] = {25.9393,-6,-6.93928}
In[7] := Eigenvectors[N[m]]
Out[7] = {{-0.120689,-0.910013,-0.396625},
          {0.159336,-0.962413,0.219937},
          {-0.19245,0.96225,-0.19245}}
In[8] := m2={{1,0,0},{0,1,1},{1,0,1}}
Out[8] = {{1,0,0},{0,1,1},{1,0,1}}
In[9] := Eigensystem[N[m2]]
```

```

Out[9] = {{1.,1.,1.},{7.34684*10-38,1.,-2.71051*10-19},
          {0.,-1.,2.71051*10-19},{0.,1.,0.}}
In[10] := Chop[%]
Out[10] = {{1.,1.,1.},{0.,1.,0.},{0.,-1.,0.},{0.,1.,0.}}

```

Команда *Eigensystem* ищет и собственные значения и собственные вектора матрицы, ответ представлен в виде набора собственных значений и соответствующих им собственных векторов  $\{\{\lambda_1, \lambda_2, \dots\}, \{\{x_1, x_2, \dots\}, \{y_1, y_2, \dots\}, \dots\}\}$ . Команда *Chop[...]* обнуляет малые добавки, возникшие как погрешности численного расчета.

Заметим, что отдельный элемент матрицы может быть получен командой  $m[[i,j]]$ , а строка —  $m[[i]]$ . Команда *Det[m]* вычисляет детерминант матрицы  $m$ , *Transpose[m]* ищет транспонированную матрицу, а командой  $Sum[m[[i,i]],\{i,Length[m]\}]$  можно вычислить след матрицы  $m$ .

При создании тензорных величин дополнительно могут использоваться команды *Table*, *Array*:

```

In[11] := Table[f[i,j],{i,2},{j,3}]
Out[11] = {{f[1,1],f[1,2],f[1,3]},
          {f[2,1],f[2,2],f[2,3]}}
In[12] := Array[f,5]
Out[12] = {f[1],f[2],f[3],f[4],f[5]}

```

*Mathematica* решает системы линейных алгебраических уравнений. Если  $m$  известная матрица, а  $b$  вектор, то команда *LinearSolve[m,b]* решает уравнение  $m.x=b$  относительно вектора  $x$ :

```

In[13] := m={{1,0,0},{0,1,1},{1,0,1}}
Out[13] = {{1,0,0},{0,1,1},{1,0,1}}
In[14] := b={3,2,1}
Out[14] = {3,2,1}
In[15] := LinearSolve[m,b]
Out[15] = {3,4,-2}

```



## 16. Символьные вычисления

Mathematica может оперировать с алгебраическими выражениями, которые содержат переменные или символы.

### 16.1. Дифференцирование и интегрирование

Общий вид команды дифференцирования функции нескольких переменных  $F(x_1, x_2, \dots)$  следующий:

$$D[F, \{x_1, n_1\}, \{x_2, n_2\}, \dots],$$

что соответствует дифференцированию функции  $F$  по переменной  $x_i$   $n_i$  раз.

```
In[1] := D[Exp[ax] (x^3+y^3), {x, 3}, {y, 2}
```

```
Out[1] = 6a E y
```

Mathematica выполняет аналитически интегрирование (включая кратные интегралы) в конечных и бесконечных пределах элементарных и специальных функций:

```
In[2] := Integrate[(1-x)^3 (1-y)^2, {y, x, 1}, {x, 0, 1}]
```

```
Out[2] = -- x x x
          12 4 4 12
```

```
In[3] := Integrate[Exp[a x^2+bx+c], {x, -Infinity, Infinity}]
```

```
Out[3] = - ( Sqrt[-a]E Sqrt[Pi]
              2 -b/(4a)+c
              ----- )
              a
```

```
In[4] := PowerExpand[%]
```

```
Out[4] = - IE Sqrt[Pi]
          2 -b/(4a)+c
          -----
          Sqrt[a]
```

In[5] := Integrate[BesselJ[3,x],{x,0,5}]

2 BesselJ[1,5]

Out[5] = 1 -  $\frac{2 \text{BesselJ}[1,5]}{5}$  - BesselJ[2,5]

In[6] := N[%]

Out[6] = 1.08447

In[7] := Integrate[BesselJ[n,x],{x,0,1}]

1  $\frac{1+n}{2}$   $\frac{3+n}{2}$

Out[7] =  $((-)^{(n/2)} \text{HypergeometricPFQ}[\{\frac{1}{4}\}, \{\frac{1+n}{2}, \frac{3+n}{2}\},$

$\frac{1}{4} - (-)] / \text{Gamma}[2+n]$

## 16.2. Ряды и пределы

Пакет **Mathematica** позволяет выполнить разложение в ряд по членам  $n$ -го порядка по степеням малого параметра  $(x-x_0)$  любых функций, заданных аналитически командой  $\text{Series}[F, \{x, x_0, n\}]$ :

In[1] := F=Exp[a x] Cos[bx+c]

ax

Out[1] = E Cos[bx+c]

In[2] := F1=Series[F,{x,0,2}]

2 2

a b 2 3

Out[2] =  $1+ax+(\frac{a^2}{2} - \frac{b^2}{2})x^2 + O[x^3]$

In[3] := Normal[F1]

2 2

a b 2

Out[3] =  $1+ax+(\frac{a^2}{2} - \frac{b^2}{2})x^2$

Команда  $\text{Normal}[\dots]$  отбрасывает слагаемое  $O[(x-x_0)^{n+1}]$  в разложении и приводит результат, полученный командой  $\text{Series}$ , к виду, который может использоваться при дальнейших преобразованиях.

**Mathematica** находит пределы аналитически заданных функций:  $\text{Limit}[\text{Func}, x \rightarrow x_0, \text{Direction} \rightarrow \pm 1]$ , где опция *Direction* определяет слева или справа берется предел.

```
In[4] := Limit[Sin[x]/x, x->0]
Out[4] = 1.
In[5] := Limit[Tan[x], x->Pi/2, Direction->1]
Out[5] = Infinity
In[6] := Limit[Tan[x], x->Pi/2, Direction->-1]
Out[6] = -Infinity
```

### 16.3. Полиномы и рациональные функции

В системе **Mathematica** есть несколько команд, которые значительно облегчают работу с полиномами:

*Expand*[poly] — разложение полинома (**Mathematica** не раскрывает скобки!),  
*Factor*[poly] — факторизация полинома (процедура, обратная *Expand*),  
*FactorTerms*[poly] — вынос за скобки общего множителя.

```
In[1] := poly=(x-a)^2 (y-b)
          2      2
Out[1] = (-a+x) (-b+y)
In[2] := Expand[poly]
          2      2      2
Out[2] = -(a b)+2abx-bx +a y-2axy+x y
In[3] := Factor[%]
          2
Out[3] = (-a+x) (-b+y)
```

*Collect*[poly, x] — представляет данное выражение как полином по степеням x,  
*PolynomialGCD*[poly1, poly2, ...] — нахождение общего делителя полиномов,  
*PowerExpand*[...] — упрощает степенные выражения типа  $\sqrt{x^2} \rightarrow x$ .

```

In[4] := Collect[poly,a]
          2  2  2
Out[4] = -(bx )+x y+a (-b+y)+a(2bx-2xy)
In[5] := poly1=a-ax-y+xy
Out[5] = a-ax-y+xy
In[6] := poly2=b-bx-y+xy
Out[6] = b-bx-y+xy
In[7] := PolynomialGCD}[poly1,poly2]
Out[7] = -1+x
In[8] := Sqrt[ab^2]/(ab)
          2
Out[8] = Sqrt[ab ]/(ab)
In[9] := PowerExpand[%]
          1
Out[9] = -----
          Sqrt[a]

```

Для работы с рациональными дробями в системе Mathematica существуют команды:

*Together[expr]* — приводит к общему знаменателю,  
*Apart[expr]* — разлагает на простые дроби.

```

In[10] := f=(a*x*y+b*x+c*y)/(x*y)
          axy+bx+cy
Out[10] =-----
          xy
In[11] := Apart[f]
          c+ax  b
Out[11] = ---- + -
          x    y
In[12] := Together[%]
          axy+bx+cy
Out[12] = -----
          xy

```

При преобразованиях полиномов может быть удобной команда *Take*, смысл которой демонстрирует данный пример:

```
In[13] := t=Series[Exp[x*y],{x,0,3}]
```

```
Out[13] = 1+xy+  $\frac{x^2 y^2}{2}$  +  $\frac{x^3 y^3}{6}$  + 0[x]^4
```

```
In[14] := t=Normal[t]
```

```
Out[14] = 1+xy+  $\frac{x^2 y^2}{2}$  +  $\frac{x^3 y^3}{6}$ 
```

```
In[15] := Take[t,3]-Take[t,2]
```

```
Out[16] =  $\frac{x^2 y^2}{2}$ 
```

## 16.4. Операции с уравнениями

**Mathematica** аналитически находит корни полинома четвертой степени с произвольными коэффициентами, хотя следует отметить, что в частных случаях решаются уравнения и более высокого порядка.

```
In[1] := Solve[ax^4+bx^3+cx^2+dx+f == 0, x]
```

Запуская ядро, получаем ответ.

Так же, как и в случае численных вычислений, **Mathematica** способна решать систему уравнений при помощи команды

$$\text{Solve}\{\{\text{eqn}_1, \text{eqn}_2, \dots, \text{eqn}_n\}, \{x_1, x_2, \dots\}\}.$$

Для удобства представления результатов используют команду *ColumnForm[expr]*, которая выводит каждое решение на отдельной строке. Следует отметить, что **Mathematica** выдает любые результаты без приведения подобных членов, поэтому имеется команда *Simplify[expr]*, которая приводит подобные члены в выражении.

**Mathematica** решает аналитически обыкновенные дифференциальные уравнения и системы уравнений:

```
In[2] := R=DSolve[{y''[x]+w^2 y[x]==a*Cos[x],
  y[0]==0,y'[0]==0},y[x],x]
      -Iwx      Iwx
      a E      a E      a Cos[x]
Out[2] = ----- + ----- + -----
          2      2      2
        2-2w    2-2w    -1+w
```

```
In[3] := Simplify[y[x]/.%]
```

```
      -I w x      2 I w x      I w x
      a E      (1 + E      - 2 E      Cos[x])
Out[3] = {-----}
              2
            2 - 2 w
```

```
In[4] := Simplify[ComplexExpand[%]]
```

```
      a (Cos[x] - Cos[w x])
Out[4] = {-----}
              2
            -1 + w
```

В рассмотренном примере решается уравнение

$$y''(x) + w^2 y(x) = a \cos(x)$$

с начальными условиями  $y(0) = 0$ ,  $y'(0) = 0$ .

Уравнение вида  $x''(t) = -kx'(t)^2$  с помощью *DSolve* не решается:

```
In[5] := eqn=x''[t]==-k*x'[t]^2};
      DSolve[eqn,x[0]==0,x'[0]==2,x[t],t]
```

```
DSolve::dnim:
Built-in procedures cannot solve this differential
equation.
```

```
Out[5] = DSolve[{x''[t]==-k*x'[t]^2,x[0]==0,x'[0]=2,x[t],t}
```

Его решение, однако, можно найти с помощью пакета *Calculus* (см. стр. 61).

При решении системы обыкновенных дифференциальных уравнений используется команда:

$$DSolve[\{eqn_1, eqn_2, \dots, eqn_n\}, \{y_1[x], y_2[x], \dots, y_n[x]\}, x].$$

## 17. Программирование и замена переменных

Для замены переменных в алгебраическом выражении используется команда  $\rightarrow$ , например,

$$\text{In}[1] := 1+x^2+3x^3/. x \rightarrow 1+a$$

которая применима и для задания новых функций

$$\begin{aligned} \text{In}[2] &:= \{f[1], f[2], f[3]\} /. f[n_] \rightarrow n^2 \\ \text{Out}[2] &= \{1, 4, 9\} \end{aligned}$$

Но есть и другая возможность описывать новые функции, например, предыдущую функцию можно задавать как

$$\begin{aligned} \text{In}[3] &:= f[n_] := n^2 \\ \text{In}[4] &:= f[x+y] \\ &\quad 2 \\ \text{Out}[4] &= (x+y)^2 \end{aligned}$$

Более сложный пример пользовательской функции-процедуры:

$$\begin{aligned} \text{In}[5] &:= F[x_] := (t=(1+x)^2; t=Expand[t]) \\ \text{In}[6] &:= F[a+b] \\ &\quad 2 \qquad \qquad \qquad 2 \\ \text{Out}[6] &= 1 + 2a + a^2 + 2b + 2ab + b^2 \end{aligned}$$

В системе *Mathematica* существует несколько возможностей организации циклов. Во-первых, при помощи команды *Do*:

$$\begin{aligned} &Do[\{expr_1, expr_2, \dots, \{i, i_{min}, i_{max}, istep\}\}, \\ &\quad Do[\{expr_1, expr_2, \dots, \{n\}\}. \end{aligned}$$

В первом случае параметр цикла  $i$  меняется от  $i_{\min}$  до  $i_{\max}$  с шагом  $i\text{step}$ , а во втором — параметр  $n$  задает число повторений.

In[7] := t=x;Do[t=1/(1+kt),{k,2,4}];t

Out[7] = 
$$\frac{1}{1 + \frac{4}{1 + \frac{3}{1 + 2x}}}$$

Оператор  $Nest[F,expr,n]$  — функция  $F$   $n$  раз действует на выражение  $expr$ .

In[8] := Nest[F,x,3]

Out[8] = F[F[F[x]]]

Оператор  $For[start, test, step, expr]$  — выполняется  $start$ , затем выполняется  $step$  и  $expr$ , пока  $test=True$ :

In[9] := For[i=1;t=x,2<i<10,,i++,t=t^2+i;Print[t]]

Out[9] = 
$$1+x$$
  

$$2+(1+x)^2$$
  

$$3+(2+(1+x)^2)^2$$

где  $i++$  эквивалентно  $i=i+1$  ( $i--$  эквивалентно  $i=i-1$ ). В системе *Mathematica* есть условные операторы:  $While[test,expr]$ , где  $test$  — проверяемое условие ( $a<,=,>b$ ),  $expr$  — выражение, которое выполняется до тех пор, пока  $test=True$  (истина) и  $If[test,then,else]$ , где, если  $test=True$ , выполняется  $then$ , а если  $test=False$  (ложь), то выполняется  $else$ .

Внутри операторов  $Do$ ,  $For$ ,  $CompoundExpression[expr_1,expr_2,...]$  могут использоваться операторы  $Goto[name]$  — переход к метке  $name$  и  $Label[name]$  — метка  $name$ .



## 18. Mathematica, TeX, Fortran, C

**Mathematica** обычно выводит полученные выражения в стандартной форме, используемой при вычислении на листе бумаги:

```
In[1] := (b^3+c^3)/(x+y)^3
          3   3
          b + c
Out[1] = -----
          3
          (x+y)
```

Но, как можно увидеть из вышесказанного, при использовании собственного редактора системы **Mathematica** все выражения представляются в виде, близком к **TeX**. Для удобства выражения могут сразу конвертироваться либо в собственный формат системы **Mathematica** командой *InputForm*{expr}

```
In[2] := InputForm[%]
Out[2] // InputForm = (a^2+b^2)/(x+y)^3
```

либо в *PlainTeX* формат

```
In[3] := TeXForm[%]
Out[3] // TeXForm={{a^2}+{b^2}}\over{{\left(x+y\right)}^3}}
```

либо в *Fortran* формат

```
In[4] := FortranForm[%]
Out[4] // FortranForm=(a**2+b**2)/(x+y)**3
```

либо в *C* формат

```
In[5] := CForm[%]
Out[5] // CForm=(Power(a,2)+Power(b,2))/Power(x+y,3)
```

Следует заметить, что вывод в формате *PlainTeX* математических формул, которые занимают несколько строк, производится в одну строку. Поэтому, после получения выражения командой *TeXForm*, необходимо вручную разбить его на отдельные строки, например:

```
In[6] := 16*m^4 + 32*m^2*M^2 + 16*M^4 + 16*m^2*s +
16*M^2*s - 16*m^2*(2*m^2 + 2*M^2 - s - t) -
16*M^2*(2*m^2 + 2*M^2 - s - t) +
8*(2*m^2 + 2*M^2 - s - t)^2 - 16*m^2*t -
16*M^2*t + 8*t^2;
```

```
In[7] := TeXForm[%]
```

```
Out[7] // TeXForm= 16\,{m^4} + 32\,{m^2}\,{M^2} +
16\,{M^4} + 16\,{m^2}\,s + 16\,{M^2}\,s -
16\,{m^2}\, \left( 2\,{m^2} + 2\,{M^2} -
s - t \right) - 16\,{M^2}\, \left( 2\,{m^2} +
2\,{M^2} - s - t \right) + 8\,{\left( 2\,{m^2} +
2\,{M^2} - s - t \right) ^2} - 16\,{m^2}\,t -
16\,{M^2}\,t + 8\,{t^2}
```

После обработки данного выражения получается длинная формула, выходящая за рамки страницы, и, только используя `\begin{eqnarray}` ... `\end{eqnarray}`, получаем следующее выражение:

$$\begin{aligned}
& 16m^4 + 32m^2M^2 + 16M^4 + 16m^2s + 16M^2s - \\
& 16m^2(2m^2 + 2M^2 - s - t) - \\
& 16M^2(2m^2 + 2M^2 - s - t) + \\
& 8(2m^2 + 2M^2 - s - t)^2 - 16m^2t - 16M^2t + 8t^2.
\end{aligned}$$

## II. Дополнительные возможности

В системе *Mathematica* создано большое количество дополнительных прикладных пакетов, используемых во многих областях знаний, например, в электротехнике, финансах, физике и даже в биологии и истории. Для того чтобы загрузить пакет, применяется команда, начинающаяся с двух знаков <<, затем нужно указать полный путь файла загрузки дополнительного пакета (обычно он называется *master.m*) и нажать комбинацию клавиш *Shift+Enter*. Например, командная строка

```
In[1] := << w:\wmath22\packages\statisti\master.m
```

загружает пакет статистической обработки данных. При загрузке пакета справа от команды появляется двойная скобка, когда загрузка заканчивается, то скобка снова становится одинарной, стандартное *Out[...]* на экран не выводится. *Mathematica* загружает пакет один раз, если пакет уже загружен, то при повторном нажатии на строке пути файла *Shift+Enter* на экране появится сообщение о том, что файл уже загружен. Поэтому, работая сразу с несколькими файлами, достаточно загрузить пакет в одном из них. При обращении к функциям незагруженного пакета система не выполняет действий, предписанных функциями, а просто повторяет обращения к ним.

Ниже будет дано краткое описание некоторых пакетов, которые могут быть полезны при проведении физических расчетов.

### 19. Пакет *Statistics*

В пакете *Statistics* определены команды статистической обработки дискретного массива данных, а также операторы основных непрерывных и дискретных статистических функций распределения вероятности, их свойства и операции над ними.

Перед началом работы с дополнительным пакетом *Statistics* загружаем его с помощью командной строки:

```
In[1] := <<...\packages\statisti\master.m
```

## 19.1. Операции со списками

Прикладные пакеты системы Mathematica обеспечивают расширение функций, управляющих списком, которые являются встроенными в нее. Дополнительные функции, полезные для управления статистическими данными, включают подсчет частоты и вычисление накопленной (полной) суммы.

Задаем список *data*:

```
In[1] := {data}={{1,2,6,3},{8,3,5,a},{8,5,3,4},{6,2,4,c}}
Out[2] = {{1,2,6,3},{8,3,5,a},{8,5,3,4},{6,2,4,c}}
```

*Column[data,n]* — дает *n*-й столбец из *data*,  
*Column[data,{n<sub>1</sub>,n<sub>2</sub>}]* — дает список из столбцов *n<sub>1</sub>*, *n<sub>2</sub>*.

```
In[2] := [Column[data,3]]
Out[2] = {6,5,3,4}
In[3] := [Column[data,{1,2}]]
Out[3] = {{1,2},{8,3},{8,5},{6,2}}
```

*ColumnTake[data,spec]* — берет точно установленные столбцы в *data*,

*ColumnDrop[data,spec]* — отправляет точно установленные столбцы в *data*,

*ColumnJoin[data1,data2]* — соединяет элементы в соответствующих рядах в *data*.

```
In[4] := [ColumnTake[data,1]]
Out[4] = {{1},{8},{8},{6}}
In[5] := data2={{6,4,2,b},{2,5,3,2},{7,2,m,5},{2,6,1,0}}
Out[5] = {{6,4,2,b},{2,5,3,2},{7,2,m,5},{2,6,1,0}}
```

```
In[6] := d=ColumnJoin[data,data2]
Out[6] = {{1,2,6,3},{8,3,5,a},{8,5,3,4},
          {6,2,4,c}, {6,4,2,b},{2,5,3,2}}
```

```
In[7] := MatrixForm[d]
```

```
Out[7] = 1 2 6 3
          8 3 5 a
          8 5 3 4
          6 2 4 c
          6 4 2 b
          2 5 3 2
```

*RowJoin*[*data*<sub>1</sub>,*data*<sub>2</sub>] — соединяет элементы в соответствующих рядах в *data*<sub>*n*</sub>.

```
In[8] := d2=RowJoin[data,data2]
Out[8] = {{1,2,6,3,6,4,2,b},{8,3,5,a,2,5,3,2},
          {8,5,3,4,7,2,m,5},{6,2,4,c,2,6,1,0}}
```

```
In[9] := MatrixForm[d2]
Out[9] = 1 2 6 3 6 4 2 b
          8 3 5 a 2 5 3 2
          8 5 3 4 7 2 m 5
          6 2 4 c 2 6 1 0
```

*DropNonNumeric*[*data*] — опускает элементы или ряды, которые содержат нечисловые элементы в *data*,

*DropNonNumericColumn*[*data*] — опускает столбцы, которые содержат нечисловые элементы в *data*.

```
In[10] := DropNonNumeric[d2]
Out[10] = {}
In[11] := DropNonNumeric[d]
Out[11] = {{1,2,6,3},{8,5,3,4},{2,5,3,2},{2,6,1,0}}
In[12] := DropNonNumericColumn[d2]
Out[12] = {{1,2,6,6,4},{8,3,5,2,5},{8,5,3,7,2},{6,2,4,2,6}}
In[13] := DropNonNumericColumn[d]
Out[13] = {{1,2},{8,3},{8,5},{6,2},{6,4},{2,5},{7,2},{2,6}}
```

*BinCounts*[*list*, {*xmin*,*xmax*,*dx*}] — дает список, элементы которого — числа попаданий элементов списка *list* в интервалы шириной *dx* между *xmax* и *xmin*,

*Frequencies*[*list*] — дает список различных элементов в списке вместе с частотами, с которыми они встречаются (на первом месте стоит частота появления числа в списке, а на втором — само число),

*CumulativeSums*[*list*] — дает полную (накопленную) сумму списка (каждое последующее число списка равно сумме предыдущих).

```

In[27] := l={2,4,3,7,3,9,2,7,2,3,4,2,3}
Out[28] = {2,4,3,7,3,9,2,7,2,3,4,2,3}
In[29] := Frequencies[l]
Out[29] = {{4,2},{4,3},{2,4},{2,7},{1,9}}
In[30] := CumulativeSums[l]
Out[31] = {2,6,9,16,19,28,30,37,39,42,46,48,51}
In[32] := list=Table[Random[],{i,100}];
In[33] := BinCounts[list,{0,3,0.2}]
Out[33] = {14,15,15,12,8,6,7,8,3,4,0,1,2,0,2}

```

## 19.2. Обработка дискретного массива данных

Дискретный массив данных задается с помощью списка  $list = \{x_1, x_2, \dots, x_n\}$ , отдельный элемент списка  $list$  может быть получен с помощью команды  $list[[i]]$ .

Пакет **Statistics** удобен при нахождении центрального момента и средних значений массива:

$CentralMoment[list, n]$	— определяет $n$ -й центральный момент массива	$\frac{\sum_i^n (x_i - \bar{x})^n}{n}$ ;
$GeometricMean[list]$	— определяет среднее геометрическое значение массива	$\sqrt[n]{x_1 x_2 \dots x_n}$ ;
$Mean[list]$	— определяет среднее арифметическое значение массива	$\frac{1}{n} \sum_i^n x_i$ ;
$MeanDeviation[list]$	— определяет среднее абсолютное отклонение данных массива от среднего	$\frac{\sum_i^n  x_i - \bar{x} }{n}$ ;

```

In[1] := list={2,6,2,5,4,3,7,5,3,6,8,4,3};
In[2] := N[CentralMoment[list,5]]
Out[2] = 36.6659
In[3] := N[Mean[list]]
Out[3] = 4.46154
In[4] := N[GeometricMean[list]]
Out[4] = 4.08101
In[5] := N[MeanDeviation[list]]
Out[5] = 1.57396

```

*Median*[list] — определяет медиану массива

*RootMeanSquare*[list] — определяет среднеквадратичный корень массива  $\sqrt{\bar{x}^2}$ ;

*SampleRange*[list] — определяет амплитуду  $x_{\max} - x_{\min}$ ;

```
In[6] := Median[list]
Out[6] = 4
In[7] := N[RootMeanSquare[list]]
Out[7] = 4.81983
In[8] := SampleRange[list]
Out[8] = 6
```

*StandardDeviation*[list] — определяет стандартное отклонение массива  $\sqrt{\frac{\sum_i^n (x_i - \bar{x})^2}{n-1}}$ ;

*StandardDeviationMLE*[list] — определяет стандартное отклонение массива с максимальной вероятностной оценкой  $\sqrt{\frac{\sum_i^n (x_i - \bar{x})^2}{n}}$ ;

*StandardErrorOfSampleMean*[list] — определяет стандартное отклонение массива от среднего значения  $x$   $\sqrt{\frac{\sum_i^n (x_i - \bar{x})^2}{n(n-1)}}$ ;

```
In[9] := N[StandardDeviation[list]]
Out[9] = 1.89804
In[10] := N[StandardDeviationMLE[list]]
Out[10] = 1.82358
In[11] := N[StandardErrorOfSampleMean[list]]
Out[11] = 0.526422
```

*Variance*[list] — определяет дисперсию данных массива  $\frac{\sum_i^n (x_i - \bar{x})^2}{n-1}$ ;

*VarianceMLE*[list] — определяет дисперсию данных массива с максимальной вероятностной оценкой  $\frac{\sum_i^n (x_i - \bar{x})^2}{n}$ ;

*VarianceOfSampleMean*[list] — определяет дисперсию среднего значения  $x$   $\frac{\sum_i^n (x_i - \bar{x})^2}{n(n-1)}$ ;

In[12] := N[Variance[list]]  
 Out[12] = 3.60256  
 In[13] := N[VarianceMLE[list]]  
 Out[13] = 3.32544  
 In[14] := N[VarianceOfSampleMean[list]]  
 Out[14] = 0.27712

### 19.3. Непрерывные распределения

<i>BetaDistribution</i> [ $\alpha, \beta$ ]	— $\beta$ -распределение с параметрами $\alpha$ и $\beta$	$p(x) = \frac{(1-x)^{\beta-1} x^{\alpha-1}}{\beta \Gamma(\alpha, \beta)}$ ;
<i>CauchyDistribution</i> [ $a, b$ ]	— распределение Коши с параметрами $a$ и $b$	$p(x) = \frac{ab}{\pi(a^2 + b^2 x^2)}$ ;
<i>ChiDistribution</i> [ $n$ ]	— распределение $\chi$ с $n$ степенями свободы	$p(x) = \frac{x^{n-2} x^{n-1}}{e^{x^2/2} \Gamma(\frac{n}{2})}$ ;
<i>ChiSquareDistribution</i> [ $n$ ]	— распределение $\chi^2$ с $n$ степенями свободы	$p(x) = \frac{x^{n/2-1}}{2^{n/2} e^{x/2} \Gamma(\frac{n}{2})}$ ;
<i>ExponentialDistribution</i> [ $a$ ]	— экспоненциальное распределение с параметром $a > 0$	$p(x) = \frac{a}{e^{ax}}$ ;
<i>GammaDistribution</i> [ $\alpha, \beta$ ]	— $\Gamma$ -распределение с параметрами $\alpha$ и $\beta$	$p(x) = \frac{x^{\alpha-1}}{\beta^\alpha e^{x/\beta} \Gamma(\alpha)}$ ;
<i>LaplaceDistribution</i> [ $\mu, \beta$ ]	— распределение Лапласа со средним значением $\mu$ и изменением параметра $\beta$	$p(x) = \frac{1}{2\beta} e^{-\frac{ x-\mu }{\beta}}$ ;
<i>NormalDistribution</i> [ $\mu, \sigma$ ]	— нормальное распределение со средним значением $\mu$ и стандартным отклонением $\sigma$	$p(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$ ;
<i>UniformDistribution</i> { $min, max$ }	— однородное распределение на интервале { $min, max$ }	$p(x) = \frac{ x-min - x-max }{2(max-min)}$ ;
<i>StudentTDistribution</i> [ $n$ ]	— распределение Стьюдента с $n$ степенями свободы	$p(x) = \frac{\Gamma(\frac{n+1}{2})}{\sqrt{n} \Gamma(\frac{n}{2})} (1+x^2)^{-\frac{n+1}{2}}$

### 19.4. Дискретные распределения

<i>BernoulliDistribution</i> [ $p$ ]	— представляет распределение Бернулли со средним числом $p$ ;
<i>BinomialDistribution</i> [ $n, p$ ]	— биномиальное распределение для $n$ испытаний с вероятностью $p$ ;



<i>DiscreteUniformDistribution</i> [n]	— дискретное однородное распределение с n возможными результатами;
<i>GeometricDistribution</i> [p]	— геометрическое распределение для вероятности p;
<i>LogSeriesDistribution</i> [θ]	— распределение логарифмического ряда с параметром θ;
<i>NegativeBinomialDistribution</i> [n,p]	— отрицательное биномиальное распределение для учтенной неудачи n и вероятности p;
<i>PoissonDistribution</i> [μ]	— распределение Пуассона со средним значением μ.

## 19.5. Операции с распределениями

В системе **Mathematica** не только определены распределения, но и заданы функции над ними.

<i>CDF</i> [distribution,x]	— определяет кумулятивную плотность вероятности заданного распределения: $CDF[\dots, x] = \int_{x_{\min}}^x PDF[\dots, x']x'$ ;
<i>Domain</i> [distribution]	— определяет область заданного распределения;
<i>Mean</i> [distribution]	— определяет среднее значение указанного распределения;
<i>PDF</i> [distribution,x]	— определяет плотность вероятности заданного распределения как функцию x.

```
In[1] := CDF[NormalDistribution[0,1],x]
```

$$1 + \frac{\text{Erf}\left[\frac{x}{\sqrt{2}}\right]}{\sqrt{2}}$$

```
Out[1] = -----
```

2

```
In[2] := Domain[NormalDistribution[0,1]]
```

```
Out[2] = { -Infinity, Infinity }
```

```
In[3] := Mean[NormalDistribution[0,1]]
```

Out[3] = 0

In[4] := PDF[NormalDistribution[0,1],x]

$$\text{Out[4]} = \frac{1}{\sqrt{2\pi}} e^{-x^2/2}$$

- Random* [distribution] — дает случайное число указанного распределения;
- StandardDeviation* [distribution] — определяет стандартное отклонение указанного статистического распределения;
- Variance* [distribution] — определяет дисперсию указанного распределения.

In[5] := Random[NormalDistribution[0,1]]

Out[5] = 1.018683125239047

In[6] := StandardDeviation[NormalDistribution[0,1]]

Out[6] = 1

In[7] := Variance[NormalDistribution[0,1]]

Out[7] = 1

## 20. Пакет Algebra

В пакете **Algebra** определены команды тригонометрических упрощений, также **Algebra** обеспечивает символьное представление сумм и произведений.

Перед началом работы с дополнительным пакетом **Algebra** загружаем его с помощью командной строки:

```
In[1] := <<...\packages\algebra\master.m
```

**Algebra** расширяет встроенные функции *Sum* и *Product* для символьных значений. Например, можно выразить символьную сумму типа  $Sum[a[k], \{k, k_{min}, k_{max}\}]$ , где  $a[k+1]/a[k]$  — рациональная функция.

```
In[2] := Sum[Sin[k], {k, a, b}]
```

```
Out[2] = Csc[-] Sin[- - - + -] Sin[- + -]
          1      1  a  b      a  b
          2      2  2  2      2  2
```

*SymbolicSum*[ $f, \{i, i_{min}, i_{max}\}$ ] — ищет значение  $Sum[f, \{i, i_{min}, i_{max}\}]$  для символов  $i_{min}, i_{max}$ ;

*SymbolicSum*[ $f, \{i, i_{max}\}$ ] — ищет сумму  $f$  по  $i$  от 1 до  $i_{max}$ .

```
In[3] := SymbolicSum[i^3, {i, n}]
```

```
Out[3] =      2      2
          n  (1 + n)
          -----
              4
```

```
In[4] := SymbolicSum[k^6, {i, 4, n}]
```

```
Out[4] = k (-3 + n)
```

```
In[5] := SymbolicSum[1/(2 k^2+1), {k, 0, Infinity}]
```

```
Out[5] = 1/2 + -----
          Pi Coth[-----]
          Sqrt[2]
```

Algebra обеспечивает тригонометрические упрощения с помощью функций *TrigReduce*, *TrigFactor*, *TrigToComplex* и *ComplexToTrig*.

*TrigExpand[expr]* = *Expand[expr, Trig -> True]*,

*TrigFactor[expr]* — записывает сумму тригонометрических функций как произведение,

*TrigReduce[expr]* — записывает тригонометрические функции многократных углов как суммы произведений тригонометрических функций этого угла (упрощает аргументы тригонометрических функций).

In[1] := TrigFactor[Cos[x]+Cos[y]]

Out[1] = 2 Cos[- - ] Cos[- + -]  

$$2 \cos\left[-\frac{x}{2} - \frac{y}{2}\right] \cos\left[-\frac{x}{2} + \frac{y}{2}\right]$$

In[2] := TrigReduce[Cos[x+y]+Sin[y-x]]

Out[2] = Cos[x] Cos[y] - Cos[y] Sin[x] + Cos[x] Sin[y] - Sin[x] Sin[y]

In[3] := TrigReduce[Sin[4x]]

Out[3] = 4 Cos[x]<sup>3</sup> Sin[x] - 4 Cos[x] Sin[x]<sup>3</sup>

*TrigToComplex[expr]* — записывает тригонометрические функции в виде комплексных экспонент,

*ComplexToTrig[expr]* — записывает комплексные экспоненты как тригонометрические функции вещественного угла.

In[4] := TrigToComplex[Cos[x]+Sin[y]]

Out[4] = 
$$\frac{e^{-I x} + e^{I x}}{2} - \frac{i}{2} (-e^{-I y} + e^{I y})$$

In[5] := ComplexToTrig[%]

Out[5] = Cos[x] + Sin[y]

In[6] := Simplify[TrigToComplex[Sin[x/2]^2 Cos[2x]]

$$\text{Out}[6] = \frac{-3Ix^3 - (1+Ex^2)(1+Ex^4)}{8}$$

## 21. Пакет Calculus

В пакете **Calculus** возможно нахождение пределов выражения, содержащих в себе элементы большого класса элементарных и специальных функций. **Calculus** расширяет встроенное *Integrate* к регулярным эллиптическим и связным интегралам. Некоторые классы неопределенных интегралов могут быть выражены в терминах эллиптических функций *EllipticF*[p,x], *Elliptic*[p,x] и *Elliptic*[n,p,x].

Перед началом работы с дополнительным пакетом **Calculus** загружаем его с помощью командной строки:

```
In[1] := <<...\packages\calculus\master.m
```

**Calculus** расширяет встроенный *DSolve*, вычисляя дифференциальные уравнения гипергеометрического типа, избранные нелинейные уравнения 1-го и 2-го порядка, избранные 2-х и 3-х мерные нелинейные системы;

```
In[2] := eqn=x'[t]==-k*x'[t]^2;
```

```
DSolve[eqn,x[0]==0,x'[0]=2,x[t],t]
```

$$\text{Out}[2] = \left\{ x[t] \rightarrow \left( \frac{\text{Log}\left[\frac{1}{2}\right]}{k} \right) + \left( \frac{\text{Log}\left[\frac{1}{2} + kt\right]}{k} \right) \right\}$$

В пакете **Calculus** возможно нахождение Паде-аппроксимации к заданной функции.

$\text{Pade}\{\text{func},\{x,x_0,m,k\}\}$  — Паде-аппроксимация функции *func* (функция переменной *x*), где *m* и *k* — степень числителя и знаменателя соответственно.

In[3] := Pade[Cos[x],{x,0,4,4}]

$$1 - \frac{115 x^2}{252} + \frac{313 x^4}{15120}$$

Out[3] = -----

$$1 + \frac{11 x^2}{252} + \frac{13 x^4}{15120}$$

## 21.1. $\delta$ -функция Дирака

В пакете *Calculus* определяется функция Хевисайда и ее обобщенная производная  $\delta$ -функция Дирака. В расширенных функциях *Limit*, *Integrate* и *Derivative* применяются функция Хевисайда и  $\delta$ -функция Дирака, как это видно из приведенных примеров.

*DiracDelta*[*x*] —  $\delta$ -функция Дирака,

*DiracDelta*[*x*<sub>1</sub>,*x*<sub>2</sub>,... ] — многомерная  $\delta$ -функция Дирака.

In[1] := Integrate[Sin[x] Exp[-x] DiracDelta[x-a], {x, 0, Infinity}]

$$\text{Out}[1] = \frac{\text{Sin}[a]}{E^a} - \frac{\text{Sin}[a] \text{UnitStep}[-a]}{E^a}$$

In[2] := Integrate[Sin[x] Exp[-x] DiracDelta'[x-a], {x, 0, Infinity}]

$$\text{Out}[2] = -\left(\frac{\text{Cos}[a] - \text{Sin}[a]}{E^a}\right) - \frac{\text{DiracDelta}[a] \text{Sin}[a]}{E^a} +$$

$(\text{Cos}[a] - \text{Sin}[a]) \text{UnitStep}[-a]$

---

a  
E

- $\text{UnitStep}[x]$  — функция Хевисайда, равная 1 для  $x > 0$  и = 0 для  $x < 0$ ,
- $\text{UnitStep}[x_1, x_2, \dots]$  — многомерная функция Хевисайда, равная 1 для  $(x_1 > 0)$  и  $(x_2 > 0)$ ... и = 0 для  $(x_1 < 0)$ ,  $(x_2 < 0)$ .

## 21.2. Преобразования Фурье

Calculus обеспечивает экспоненциальные и тригонометрические Фурье-преобразования и ряды.

- $\text{FourierTransform}[expr, t, w]$  — одномерное преобразование Фурье,
- $\text{InverseFourierTransform}[expr, w, t]$  — обратное одномерное преобразование Фурье,
- $\text{NFourierTransform}[expr, t, w]$  — численное значение одномерного преобразования Фурье.

In[1] :=  $\text{FourierTransform}[t^2 \text{Exp}[-2 t], t, w]$

Out[1] =  $-2 \text{Pi DiracDelta}'[-2 I - w]$

In[2] :=  $\text{InverseFourierTransform}[\%, w, t]$

2  
t  
Out[2] =  $\frac{t^2}{2 t}$   
E

- $\text{FourierCosTransform}[expr, t, w]$  — одномерное косинус-фурье преобразование,
- $\text{InverseFourierCosTransform}[expr, w, t]$  — обратное одномерное косинус-фурье преобразование.

In[3] := FourierCosTransform[t^2 Exp[-2 t],t,w]

Out[3] = 
$$-16 \left(1 - \frac{3w}{4}\right)$$

-----  
$$(4 + w)^2$$

In[4] := InverseFourierCosTransform[%,w,t]

Out[4] = 
$$\frac{t^2}{E}$$

*FourierSinTransform*[expr,t,w]

— одномерное синус-фурье преобразование,

*InverseFourierSinTransform*[expr,w,t]

— обратное одномерное синус-фурье преобразование.

In[5] := FourierSinTransform[t^2 Exp[-2 t],t,w]

Out[5] = 
$$2 \sin\left[3 \arctan\left[\frac{w}{2}\right]\right]$$

-----  
$$(4 + w)^{3/2}$$

In[6] := InverseFourierSinTransform[%,w,t]

Out[6] = 
$$\frac{t^2}{E}$$

*FourierTrigSeries*[expr,{x,x0,x1},n]

— разложение в тригонометрический ряд n-го порядка,

*FourierExpSeries*[expr,{x,x0,x1},n]

— разложение в экспоненциальный ряд n-го порядка.



In[7] := FourierTrigSeries[x,{x,0,1},3]

$$\text{Out[7]} = \frac{1}{2} \frac{\sin[2 \text{ Pi } x]}{\text{Pi}} - \frac{\sin[4 \text{ Pi } x]}{2 \text{ Pi}} + \frac{\sin[6 \text{ Pi } x]}{3 \text{ Pi}}$$

In[8] := FourierExpSeries[x,{x,0,1},3]

$$\text{Out[8]} = \frac{1}{2} \frac{e^{-2 \text{ I Pi } x} - e^{2 \text{ I Pi } x}}{\text{Pi}} + \frac{1}{2} \frac{e^{-4 \text{ I Pi } x} - e^{4 \text{ I Pi } x}}{\text{Pi}} + \frac{1}{4} \frac{e^{-6 \text{ I Pi } x} - e^{6 \text{ I Pi } x}}{\text{Pi}} + \frac{1}{6} \frac{e^{-6 \text{ I Pi } x} - e^{6 \text{ I Pi } x}}{\text{Pi}}$$

*NFourierTrigSeries[...]* = *N[FourierTrigSeries[...]]*,  
*NFourierExpSeries[...]* = *N[FourierExpSeries[...]]*.

In[9] := NFourierTrigSeries[x,{x,0,1},3]

$$\text{Out[9]} = 0.5 - 6.93889 \cdot 10^{-17} \cos[2 \text{ Pi } x] - 5.63785 \cdot 10^{-17} \cos[4 \text{ Pi } x] - 9.02056 \cdot 10^{-17} \cos[6 \text{ Pi } x] - 0.31831 \sin[2 \text{ Pi } x] - 0.159155 \sin[4 \text{ Pi } x] - 0.106103 \sin[6 \text{ Pi } x]$$

In[10] := NFourierExpSeries[x,{x,0,1},3]

$$\text{Out[10]} = 0.5 + 0. \text{ I } +$$

$$\begin{aligned}
& (-4.16334 \cdot 10^{-17} - 0.159155 \cdot I) E^{-2 I \text{ Pi } x} + \\
& (-4.16334 \cdot 10^{-17} + 0.159155 \cdot I) E^{2 I \text{ Pi } x} + \\
& (-6.07153 \cdot 10^{-17} - 0.0795775 \cdot I) E^{-4 I \text{ Pi } x} + \\
& (-6.07153 \cdot 10^{-17} + 0.0795775 \cdot I) E^{4 I \text{ Pi } x} + \\
& (-1.57209 \cdot 10^{-17} - 0.0530516 \cdot I) E^{-6 I \text{ Pi } x} + \\
& (-1.57209 \cdot 10^{-17} + 0.0530516 \cdot I) E^{6 I \text{ Pi } x}
\end{aligned}$$

### 21.3. Преобразования Лапласа

*LaplaceTransform*[*expr*,*t*,*s*,*Options*] — прямое преобразование Лапласа,  
*InverseLaplaceTransform*[*expr*,*s*,*t*,*Options*] — обратное преобразование Лапласа.

In[1] := LaplaceTransform[t^2 Exp[-2t], t, s]

Out[1] = 
$$\frac{2}{(2+s)^3}$$

In[2] := InverseLaplaceTransform[%, s, t]

Out[2] = 
$$\frac{t^2}{E^{2t}}$$

## 21.4. Векторный анализ

Пакет **Calculus** представляет стандартные векторные дифференциальные операторы в ортогональных системах координат. Предусмотрены координатные преобразования. К сожалению, у **Calculus** есть ограничения: он работает только с праввинтовой координатной системой, поддерживает только трехмерные системы, в некоторых из координатных систем параметры и координаты не могут иметь комплексных значений.

In[1] := v1={x1,y1,z1}

Out[1] = {x1,y1,z1}

In[2] := v2={x2,y2,z2}

Out[2] = {x2,y2,z2}

In[3] := v3={x3,y3,z3}

Out[3] = {x3,y3,z3}

*DotProduct*{v<sub>1</sub>,v<sub>2</sub>} — скалярное произведение векторов v<sub>1</sub> и v<sub>2</sub> в 3-х мерном пространстве в заданной по умолчанию системе координат,

*DotProduct*{v<sub>1</sub>,v<sub>2</sub>,*Coordsys*} — скалярное произведение векторов v<sub>1</sub> и v<sub>2</sub> в 3-х мерном пространстве в системе координат *Coordsys*.

In[3] := DotProduct[v1,v2]

Out[3] = x1 x2 + y1 y2 + z1 z2

In[4] := DotProduct[v1,v2,Spherical]

Out[4] = x1 x2 Cos[y1] Cos[y2] +  
x1 x2 Cos[z1] Cos[z2] Sin[y1] Sin[y2] +  
x1 x2 Sin[y1] Sin[y2] Sin[z1] Sin[z2]

*CrossProduct*{v<sub>1</sub>,v<sub>2</sub>} — векторное произведение векторов v<sub>1</sub> и v<sub>2</sub> в 3-х мерном пространстве в заданной по умолчанию системе координат,

*CrossProduct*{v<sub>1</sub>,v<sub>2</sub>,*Coordsys*} — векторное произведение векторов v<sub>1</sub> и v<sub>2</sub> в 3-х мерном пространстве в системе координат *Coordsys*.

In[5] := v3=CrossProduct[v1,v2]

Out[5] = {-y2 z1+y1 z2,x2 z1-x1 z2,-x2 y1+x1 y2}

In[6] := CrossProduct[v1,v2,Cylindrical]

Out[6] = {Sqrt[(-(x1 z2 Cos[y1]) + x2 z1 Cos[y2])<sup>2</sup> +

(x1 z2 Sin[y1] - x2 z1 Sin[y2])<sup>2</sup> ],

ArcTan[x1 z2 Sin[y1] - x2 z1 Sin[y2],

-(x1 z2 Cos[y1]) + x2 z1 Cos[y2]],

-(x1 x2 Cos[y2] Sin[y1]) + x1 x2 Cos[y1] Sin[y2]}

*ScalarTripleProduct*{v1,v2,v3}

— смешанное произведение векторов v<sub>1</sub>, v<sub>2</sub> и v<sub>3</sub> в 3-х мерном пространстве в заданной по умолчанию системе координат,

*ScalarTripleProduct*{v1,v2,v3,Coordsys}

— смешанное произведение векторов v<sub>1</sub> и v<sub>2</sub> в 3-х мерном пространстве в системе координат *Coordsys*.

In[7] := ScalarTripleProduct[v1,v2,v3,Cartesian]

Out[7] = -(x3 y2 z1) + x2 y3 z1 + x3 y1 z2 - x1 y3 z2 -  
x2 y1 z3 + x1 y2 z3

*Cartesian*{x,y,z},

*Coordsys* может принимать следующие значения: *Cylindrical*{r,theta,z},  
*Spherical*{r,theta,phi}.

*SetCoordinates*{*Coordsys*}

— заменяет заданную по умолчанию систему координат на систему координат *Coordsys* с заданными по умолчанию именами переменных.

*SetCoordinates*[*Coordsys*,*x,y,z*] — заменяет заданную по умолчанию систему координат на систему координат *Coordsys* с именами переменных *x,y,z*.

```
In[8] := SetCoordinates[Spherical]
        DotProduct[v1,v2]
```

```
Out[8] = Spherical[r, theta, phi]
```

```
Out[9] = x1 x2 Cos[y1] Cos[y2] +
        x1 x2 Cos[z1] Cos[z2] Sin[y1] Sin[y2] +
        x1 x2 Sin[y1] Sin[y2] Sin[z1] Sin[z2]
```

*CoordinateSystem* — показывает имя заданной координатной системы,

*Coordinates*[*Coordsys*] — дает список заданных по умолчанию имен координатных переменных в системе координат *Coordsys*.

```
In[10] := CoordinateSystem
```

```
Out[10] = Spherical
```

```
In[11] := Coordinates[Cylindrical]
```

```
Out[11] = {r, theta, z}
```

*Grad*[*f*] — градиент скалярной функции *f* в трехмерном пространстве в заданной по умолчанию системе координат,

*Grad*[*f*,*Coordsys*] — градиент скалярной функции *f* в трехмерном пространстве в системе координат *Coordsys*.

```
In[12] := Grad[r^2 Cos[theta], Spherical]
```

```
Out[12] = {2r Cos[theta], -r Sin[theta], 0}
```

*Div*[*f*] — дивергенция векторной функции *f* в заданной по умолчанию системе координат,

*Div*[*f*,*Coordsys*] — дивергенция векторной функции *f* в системе координат *Coordsys*,

In[13] := Div[x,y^2,z^3]

Out[13] = 1+ 2 y + 3 z

In[14] := Div[x,y^2,z^3,Spherical]

Out[14] = 
$$\frac{\text{Csc}[\text{theta}] (r y \text{Cos}[\text{theta}] + 2 r x \text{Sin}[\text{theta}])}{r^2}$$

*Curl[f]* — ротор векторной функции *f* в заданной по умолчанию системе координат,

*Curl[f,Coordsys]* — ротор векторной функции *f* в системе координат *Coordsys*.

In[14] := Curl[x/y,y/z,z/x]

Out[14] = 
$$\left\{ \frac{y}{z}, \frac{z}{x}, \frac{x}{y} \right\}$$

*Laplacian[f]* — лапласиан скалярной функции *f* в заданной по умолчанию системе координат,

*Laplacian[f,Coordsys]* — лапласиан скалярной функции *f* в системе координат *Coordsys*.

In[15] := Laplacian[x,y^2,z^3]

Out[15] = 2 + 6 z

## 22. Пакет Graphics

Пакет подключается с помощью командной строки

```
In[1]:= <<... \packages \graphics \master.m
```

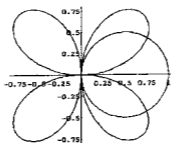
Этот пакет обеспечивает специальные функции для построения, включая гистограммы, полярные графики и полосы ошибок графиков, графики в сферических и цилиндрических координатах, 3-х мерные параметрические графики.

### 22.1. Графика в различных системах координат

Mathematica строит графики в полярной системе координат:

- PolarPlot*[ $r, \{t, t_{min}, t_{max}\}$ ] — генерирует полярный график  $r$  как функцию от  $t$ ,
- PolarPlot*[\{ $r_1, r_2, \dots$ \}, \{ $t, t_{min}, t_{max}$ \}] — строит каждое из  $r_i$  как функцию от  $t$  на том же графике,
- PolarListPlot*[\{ $r_1, r_2, \dots$ \}] — генерирует полярный график, считая, что  $r_i$  равномерно распределено по углу.

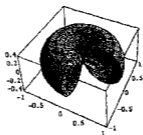
```
In[3] := PolarPlot[{Cos[t], Sin[2 t]}, {t, 0, 2 Pi}]
```



```
Out[3] = - Graphics -
```

- SphericalPlot3D*[ $r, \{\theta, \theta_{min}, \theta_{max}\}, \{\phi, \phi_{min}, \phi_{max}\}, (Opts)$ ] — рисует  $r$  как функцию от углов  $\theta$  и  $\phi$ .

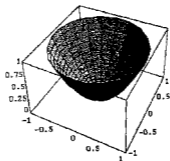
```
In[4] := SphericalPlot3D[Sin[theta]^2, {theta, 0, Pi},  
  {phi, 0, 3 Pi 2}]
```



```
Out[4] = - Graphics -
```

*CylindricalPlot3D*[ $z$ , { $r$ ,  $r_{min}$ ,  $r_{max}$ }, — рисует  $z$  как функцию  
{ $\phi$ ,  $\phi_{min}$ ,  $\phi_{max}$ }, (*Opts*)] от углов  $r$  и  $\phi$ .

```
In[5] := CylindricalPlot3D[r^2, {r, 0, 1}, {phi, 0, 2*Pi}]
```



```
Out[5] = - Graphics -
```

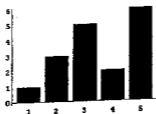
## 22.2. Гистограммы

Для построения двумерных гистограмм в системе Mathematica есть специальные функции:

*BarChart*{ $list_1, list_2, \dots$ } — генерирует гистограмму данных в списках.

```
In[1] := BarChart[{1, 3, 5, 2, 6}]
```

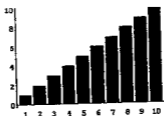




Out[1] = - Graphics -

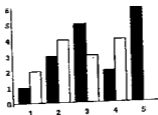
Гистограмму можно построить с помощью таблицы, что очень удобно при обработке экспериментальных данных.

In[2] := BarChart[Table[i, {i, 1, 10}]]



Out[2] = - Graphics -

In[3] := BarChart[{1, 3, 5, 2, 6}, {2, 4, 3, 4}]



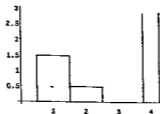
Out[3] = - Graphics -

*GeneralizedBarChart*

{{{pos<sub>1</sub>, height<sub>1</sub>, width<sub>1</sub>},  
{ pos<sub>2</sub>, height<sub>2</sub>, width<sub>2</sub>}, ...}}

— генерирует гистограмму с полосами с указанными положениями, длинами и высотами.

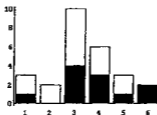
In[4] := GeneralizedBarChart[{{1, 1.5, 1}, {2, 0.5, 1}, {4, 3, 0.5}}]



Out[4] = - Graphics -

*StackedBarChart*[*list*<sub>1</sub>, *list*<sub>2</sub>, ...] — генерирует сложенную гистограмму данных в списках.

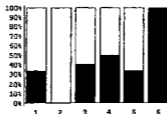
In[5] := *StackedBarChart*[{1,0,4,3,1,2},{2,2,6,3,2}]



Out[5] = - Graphics -

*PercentileBarChart* — генерирует сложенную гистограмму данных в списках такую, что сумма абсолютных значений в данной точке равна единице.

In[6] := *PercentileBarChart* [{1,0,4,3,1,2},{2,2,6,3,2}]



Out[6] = - Graphics -

Mathematica строит трехмерные гистограммы:

`BarChart3D[list, Opts]`

— создает 3-х мерную гистограмму прямоугольной матрицы `list`,

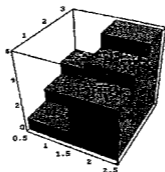
`BarChart3D[{{{z,style},...},...}, Opts]`

— создает гистограмму с определенным стилем для каждой полосы,

`BarChart3D[{{{x,y,z},style}...}, Opts]`

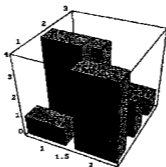
— создает гистограмму из отдельных полос с определенными `x` и `y` координатами, с высотой `z` и определенным `style`.

`In[7] := BarChart3D[{{1,2,3},{4,5,6}}]`



`Out[7] = - Graphics -`

`In[8] := BarChart3D[{{1,4,3},{4,2,1}}, XSpacing->0.2, YSpacing->0.5]`



`Out[8] = - Graphics -`

*XSpacing*, *YSpacing* — опции для *BarChart3D*, которые определяют величину пространства между полосами соответственно в *x* и *y* направлениях.

### 22.3. Списки

*PiScale*[ $x_{min}, x_{max}$ ] — дает список значений между  $x_{min}$  и  $x_{max}$ , кратных  $\pi$ ,

*PiScale*[ $x_{min}, x_{max}, n$ ] — дает список  $n$  значений между  $x_{min}$  и  $x_{max}$ , кратных  $\pi$ .

In[1] := *PiScale*[0,10].

$\pi$

Out[1] = {{0,0},{1.5708,--},{ 3.14159, $\pi$ },

2

3  $\pi$

{4.71239,----},{6.28319,2  $\pi$ },

2

5  $\pi$

{7.85398,-----},{9.42478,3  $\pi$ }}

2

In[2] := *PiScale*[0,10,4]

Out[2] = {{0,0},{3.14159, $\pi$ },{6.28319,2  $\pi$ },

{9.42478,3  $\pi$ }}

В системе *Mathematica* возможно поточечное построение двумерных графиков с меткой в точке.

*TextListPlot*{{ $y_1, y_2, \dots$ }}

— строит список с каждой точкой  $\{i, y_i\}$ , представленной как номер ее индекса  $i$ ,

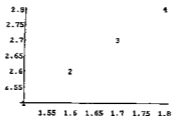
*TextListPlot*{{ $\{x_1, y_1\}, \{x_2, y_2\}, \dots$ }}

— изображает точку  $\{x_i, y_i\}$  как номер ее индекса  $i$ ,

*TextListPlot*{{ $\{x_1, y_1, t_1\}, \{x_2, y_2, t_2\}, \dots$ }}

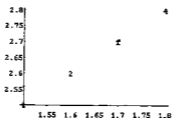
— изображает точку  $\{x_i, y_i, t_i\}$  как текст  $t_i$ .

```
In[3] := TextListPlot[{{1.5,2.5},{1.6,2.6},{1.7,2.7},
{1.8,2.8}}]
```



```
Out[3] = - Graphics -
```

```
In[4] := TextListPlot[{{1.5,2.5,t},{1.6,2.6,2},{1.7,2.7,f},
{1.8,2.8,4}}]
```



```
Out[4] = - Graphics -
```

*LabeledListPlot*{ $\{y_1, y_2, \dots\}$ }

— строит список с каждой точкой  $\{i, y_i\}$ , помеченной номером ее индекса  $i$ ,

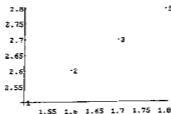
*LabeledListPlot*{ $\{x_1, y_1\}, \{x_2, y_2\}, \dots\}$ }

— помечает точку  $\{x_i, y_i\}$  номером ее индекса  $i$ ,

*LabeledListPlot* <sub>$t_i$</sub> { $\{x_1, y_1, t_1\}, \{x_2, y_2, t_2\}, \dots\}$ }

— помечает точку  $\{x_i, y_i, t_i\}$  текстом  $t_i$ .

```
In[5] := LabeledListPlot[{{1.5,2.5,1},{1.6,2.6,2},{1.7,2.7,3},
{1.8,2.8,5}}]
```



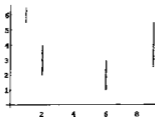
```
Out[5] = - Graphics -
```

Mathematica не только может построить график, но и показать интервал ошибки, для этого существуют специальные функции:

$ErrorListPlot[\{ \{y_1, dy_1\}, \{y_2, dy_2\}, \dots \}]$  — строит список данных с полосами ошибки в координатах  $\{i, y_i\}$ ,

$ErrorListPlot[\{ \{x_1, y_1, dy_1\}, \dots \}]$  — строит список данных с полосами ошибки в координатах  $\{x_i, y_i\}$ .

```
In[6] := ErrorListPlot[{{2,3,0.1},{9,4,0.5},{0,0,0},  
  {1,6,0.05},{6,2,0.3}}]
```



```
Out[6] = - Graphics -
```

## 23. Пакет FeynCalc

**FeynCalc** — пакет, работающий под управлением системы **Mathematica**, для проведения алгебраических вычислений в физике высоких энергий и квантовой теории поля разработан Рольфом Мертигом (<http://www.feyncalc.com>). Наряду со стандартными для системы **Mathematica** возможностями пользователю предоставляется большой набор специализированных функций и команд, ориентированных на вычисления в тензорной алгебре, преобразования и упрощения тензорных интегралов, приведение полиномов в амплитудах процессов, преобразования специальных функций. Полученные результаты могут быть конвертированы в оптимизированный файл в формате *Fortran* или *C* для дальнейших численных расчетов. **FeynCalc** может быть использован в режиме мощного научного калькулятора для упрощения произведений матриц Дирака или Гелл-Манна и спиноров, свертки громоздких выражений по тензорным индексам, вычисления следов матриц Дирака или Гелл-Манна. Ниже описаны возможности первой версии **FeynCalc 1.0**, работающей под управлением системы **Mathematica 2.2**. Последующие версии **FeynCalc (2,3)** работают только под управлением **Mathematica 3.0** или **Mathematica 4**. В них появились некоторые дополнительные функции и изменен синтаксис ряда команд, однако ядро пакета сохранено.

### 23.1. Ввод функций

#### 23.1.1. Тензоры и скалярные произведения

Основные функции:

<i>FourVector</i> [ <i>p, mu</i> ]	— 4-вектор $p_\mu$ ;
<i>LeviCivita</i> [ <i>mu, nu, ro, si</i> ]	— тензор Леви-Чевиты $\epsilon_{\mu\nu\rho\sigma}$ ;
<i>MetricTensor</i> [ <i>mu, nu</i> ]	— метрический тензор $g_{\mu\nu}$ ;
<i>ScalarProduct</i> [ <i>p, q</i> ]	— $p \cdot q$ ;
<i>Momentum</i> [ <i>p</i> ]	— объявление $p$ 4-вектором.

Тензор Леви-Чевиты и вектора поляризации определены в программе **FeynCalc** в пространстве четырех измерений. Функции *FourVector*, *MetricTensor* и *ScalarProduct* могут быть определены в пространстве произвольной размерности при помощи опции *Dimension*.

Следующие примеры демонстрируют синтаксис ввода тензоров и скалярных произведений, а также формат их представления в FeynCalc .

Скалярное произведение $p, q$	In[1]:= ScalarProduct[p,q] Out[1]=p.q
Метрический тензор $g_{\mu\nu}$	In[2]:= MetricTensor[mu,nu] Out[2]=g[mu,nu]
Метрический тензор $g_{\mu\nu}$ в $D$ -мерном пространстве	In[3]:=MetricTensor[mu,nu, Dimension ->D] Out[3]=g <sub>D</sub> [mu,nu]
При $D=4$ $g_{\mu}^{\mu} = 4$	In[4]:=MetricTensor[mu,mu] Out[4]=4
При произвольном $D$ $g_{\mu}^{\mu}=D$	In[5]:=MetricTensor[mu,mu, Dimension ->D] Out[5]=D
Ввод 4-вектора $p_{\mu}$	In[6]:=FourVector[p,mu] Out[6]=p[mu]
Ввод линейной комбинации 4-векторов $(p - 2q)_{\mu}$	In[7]:=FourVector[p-2q,mu] Out[7]=(p-2q)[mu]
Вектор в $D$ -мерном пространстве	In[8]:=FourVector[p,al, Dimension ->D] Out[8]=p <sub>D</sub> [al]
Вектор поляризации $\varepsilon_{\mu}(k)$ вводится специальной командой	In[9]:=PolarizationVector[k,mu] Out[9]=ep[k,mu]
Сопряженный вектор поляризации $\varepsilon_{\mu}^*$	In[10]:= Conjugate[ PolarizationVector[k,mu]] Out[10]=ep(*)[k,mu]

Для вектора поляризации автоматически выполняется условие поперечности  $\varepsilon(k).k = 0$ .

Суммирование по поляризациям векторных частиц выполняется при помощи команды *PolarizationSum*:

<i>PolarizationSum</i> [mu,nu]	$\rightarrow -g^{\mu\nu}$ ;
<i>PolarizationSum</i> [mu,nu,k]	$\rightarrow -g^{\mu\nu} + k^{\mu}k^{\nu}/k^2$ ;
<i>PolarizationSum</i> [mu,nu,k,n]	$\rightarrow -g^{\mu\nu} - k^{\mu}k^{\nu}n^2/(k.n)^2 + (n^{\mu}k^{\nu} + n^{\nu}k^{\mu})/(k.n)$ ;



$$\Sigma \varepsilon_{\mu} \varepsilon_{\nu}^* = -g_{\mu\nu} + k_{\mu} k_{\nu} / k^2$$

$$\text{In}[1] := \text{PolarizationSum}[\mu, \nu, k]$$

$$\text{Out}[1] = -g[\mu, \nu] + \frac{k[\mu] k[\nu]}{k.k}$$

Если калибровочный вектор  $n = p_1 - p_2$

$$\text{In}[2] := \text{PolarizationSum}[\mu, \nu, k, p_1 - p_2]$$

$$\text{Out}[2] = -g[\mu, \nu] + \frac{p_1.p_1 - 2p_1.p_2 + p_2.p_2}{(k.p_1 - k.p_2)^2} k[\mu] k[\nu] + \frac{k[\nu] (p_1 - p_2)[\mu] + k[\mu] (p_1 - p_2)[\nu]}{k.p_1 - k.p_2}$$

### 23.1.2. Матрицы Дирака

Основные функции:

$$\text{DiracMatrix}[\mu, \nu, \dots] \rightarrow \text{матрицы Дирака } \gamma_{\mu} \gamma_{\nu} \dots;$$

$$\text{DiracSlash}[p, q, \dots] \rightarrow \text{матрицы } \hat{p} \hat{q} \dots;$$

$$\text{DiracMatrix}[5] \rightarrow \gamma_5;$$

$$\text{DiracMatrix}[6] \rightarrow (1 + \gamma_5)/2;$$

$$\text{DiracMatrix}[7] \rightarrow (1 - \gamma_5)/2.$$

Итак, матрица  $\gamma_{\mu}$  представлена как  $\text{DiracMatrix}[\mu]$ , свертка  $\hat{p} = p^{\mu} \gamma_{\mu}$  — как  $\text{DiracSlash}[p]$ .

В системе **Mathematica** символ “.” используется как разделитель некоммутирующих множителей, среди которых функции **FeynCalc**: *DiracMatrix*, *DiracSlash*, *Spinor*, *LeptonSpinor*, *QuarkSpinor* и *GellMannMatrix*. Символ “.” может быть также использован вместо символа “,” как разделитель при вводе функций *DiracMatrix*, *DiracSlash*, *GellMannMatrix*. В формате вывода **FeynCalc** символ некоммутативного умножения “.” опускается.

Ввод  $(\hat{p} + \hat{q} + m) \gamma_{\mu}$

$$\text{In}[1] := (\text{DiracSlash}[p+q] + m) . \text{DiracMatrix}[\mu]$$

$$\text{Out}[1] = (gs[p+q] + m) ga[\mu]$$

Ввод  $\gamma_{\rho} \gamma_5 \gamma_{\mu} \gamma_6 \gamma_a \gamma_7$

$$\text{In}[2] := \text{DiracMatrix}[ro, 5, \mu, 6, a, 1, 7]$$

$$\text{Out}[2] = ga[ro] ga[5] ga[\mu] ga[6] ga[a] ga[7]$$

Ввод  $2\hat{a}\hat{b}(\hat{d} - \hat{c})(6\hat{q} - 3\hat{p})$

$$\text{In}[3] := \text{DiracSlash}[2a.b.(d-c).(6q-3p)]$$

$$\text{Out}[3] = 6gs[a]gs[b]gs[c-d]gs[p-2q]$$

Ввод  $\hat{\varepsilon}(k) = \varepsilon^{\mu}(k) \gamma_{\mu}$

$$\text{In}[4] := \text{DiracSlash}[\text{Polarization}[k]]$$

$$\text{Out}[4] = gs[ep[k]]$$

### 23.1.3. Матрицы Гелл-Манна

Пакет `FeynCalc` позволяет проводить вычисления в квантовой хромодинамике (КХД). В данном разделе описывается ввод матриц Гелл-Манна и структурных констант группы  $SU_C(3)$ . Основные функции:

`GellMannMatrix[a,b,...]` → матрицы Гелл-Манна  $\lambda_a \lambda_b \dots$ ;  
`SU3Delta[a,b]` → символ Кронеккера  $\delta_{ab}$  для цветовых индексов  $a, b$ ;  
`SU3F[a,b,c]` → структурная константа  $f_{abc}$  группы  $SU(3)$

Некоторые примеры:

```
Ввод  $\lambda_a \lambda_b$            In[1]:=GellMannMatrix[a].
                        GellMannMatrix[b]
                        Out[1]=la[a]la[b]
Ввод  $\lambda_c \lambda_b \lambda_a \delta_{de}$ 
                        In[2]:=GellMannMatrix[c,b,a]SU3Delta[d,e]
                        Out[2]=d[d,e]la[c]la[b]la[a]
 $f_{abc} = i/4[Tr(\lambda_a \lambda_c \lambda_b) - Tr(\lambda_a \lambda_b \lambda_c)]$ 
                        In[3]:=SU3F[a,b,c]
                        Out[3]=I/4(-tr[la[a]la[b]la[c]]+
                        tr[la[a]la[c]la[b]])
 $f_{acb} = -f_{abc}$ 
                        In[4]:=SU3F[a,c,d,SU3FToTraces
                        - > False]
                        Out[4]=-f[a,b,c]
 $f_{aab} = 0$ 
                        In[5]:=SU3F[a,a,b]
                        Out[5]=0
```

Символ "`Tr[...]`" обозначает в `FeynCalc` след произведения матриц Гелл-Манна или Дирака.

### 23.1.4. Спиноры

В `FeynCalc` используется два типа спиноров. Для лептонов используются символы `LeptonSpinor` или `Spinor`. В случае кваркового спинора используется обозначение `QuarkSpinor`. Отличие `QuarkSpinor` от `LeptonSpinor` состоит в том, что матрицы Гелл-Манна не коммутируют с `QuarkSpinor`, который несет цветовой индекс.

Основные функции:

<i>Spinor</i> [p,m]	→ $u(p, m)$ или $\bar{u}(p, m)$ для лептонов;
<i>Spinor</i> [-p,m]	→ $v(p, m)$ или $\bar{v}(p, m)$ для лептонов;
<i>LeptonSpinor</i> [p,m]	→ $u(p, m)$ или $\bar{u}(p, m)$ для лептонов;
<i>LeptonSpinor</i> [-p,m]	→ $v(p, m)$ или $\bar{v}(p, m)$ для лептонов;
<i>QuarkSpinor</i> [p,m]	→ $u(p, m)$ или $\bar{u}(p, m)$ для кварков;
<i>QuarkSpinor</i> [-p,m]	→ $v(p, m)$ или $\bar{v}(p, m)$ для кварков.

Различие между спинорами  $u, v$  и  $\bar{u}, \bar{v}$  обусловлено местом спинора в амплитуде и знаком аргумента 4-импульса.

Ввод $u(p, m)$ или $\bar{u}(p, m)$	In[1]:=Spinor[p,m] Out[1]=u[p,m]
Ввод $v(p, m)$ или $\bar{v}(p, m)$	In[1]:=Spinor[-p,m] Out[1]=v[p,m]
Ввод $\bar{u}(p, m)\hat{p}$	In[2]:=Spinor[p,m].DiracSlash[p] Out[2]=u[p,m]gs[p]
Ввод $\hat{p}u(p, m)$	In[3]:=DiracSlash[p].Spinor[p,m] Out[3]=gs[p]u[p,m]
Ввод $u(p, 0)$ или $\bar{u}(p, 0)$	In[4]:=Spinor[p,0] Out[4]=u[p]

### 23.1.5. Пропагаторы

В **FeynCalc** знаменатели пропагаторов вводятся специальным образом:

$$\frac{\text{FeynAmpDenominator}[\text{PropagatorDenominator}[q,m1], \text{PropagatorDenominator}[q+p,m2], \dots]}{1} \rightarrow \frac{1}{((q^2 - m_1^2)[(q+p)^2 - m_2^2] \dots)}$$

Пример:

Ввод выражения	In[1]:=FeynAmpDenominator[ PropagatorDenominator[q,m1], PropagatorDenominator[q+p,m2]]
$1/((q^2 - m_1^2)[(q+p)^2 - m_2^2])$	Out[2]= $\frac{1}{(q^2 - m_1^2)((p+q)^2 - m_2^2)}$

## 23.2. Элементарные вычисления

### 23.2.1. Свертка по лоренцевским индексам

*Contract*{expr} → свертка по парным лоренцевским индексам.

Свернуть $g_{\alpha\beta}p^\beta$	In[1]:=Contract[MetricTensor[a1,be] FourVector[p,be]] Out[1]=p[a1]
Свернуть $q_\alpha(p-q)^\alpha$	In[2]:=Contract[FourVector[q,a1] FourVector[p-q,a1]] Out[2]=(p-q).q
Свернуть $g_{\alpha\beta}\gamma^\beta$	In[3]:=Contract[MetricTensor[a1,be] DiracMatrix[be]] Out[3]=ga[a1]
Свернуть $q_\alpha\gamma^\alpha$	In[4]:=Contract[FourVector[q,a1] DiracMatrix[a1]] Out[4]=gs[q]
Свернуть $\varepsilon_{\mu\nu\rho\sigma}p^\sigma$	In[5]:=Contract[LeviCivita[mu,nu, ro,si]FourVector[p,si]] Out[5]=eps[mu,nu,ro,p]
$\varepsilon_{\alpha\nu\rho\sigma}\varepsilon^{\beta\nu\rho\sigma} = -6g_\alpha^\beta$	In[5]:=Contract[LeviCivita[a1,nu,ro,si] LeviCivita[be,nu,ro,si], EpsContract -> True Out[5]=-6g[a1,be]
$k_\mu\varepsilon^\mu(k) = 0$	In[6]:=Contract[FourVector[k,mu] PolarizationVector[k,mu]] Out[6]=0

### 23.2.2. Упрощение скалярных произведений 4-векторов

*ExpandScalarProduct*{expr} → разложение и упрощение скалярного произведения 4-векторов

$(a+b).(c-2d)$	In[1]:=ExpandScalarProduct [ScalarProduct[a+b,c-2d]] Out[1]=a.c-2a.d+b.c-2b.d
$q_\alpha(p-q)^\alpha$	In[2]:=Contract[FourVector[q,a1] FourVector[p-q,a1]]

	Out [2]=(p-q) . q
	In [3]:=ExpandScalarProduct [%]
	Out [3]=p . q-q . q
Подстановка $q^2 \rightarrow 0$	In [4]:= % /. ScalarProduct [q, q] $\rightarrow 0$
	Out [4]=p . q
Разложить $(a + b)_\mu$	In [5]:=ExpandScalarProduct [FourVector [a+b, mu]]
	Out [5]=a [mu]+b [mu]

### 23.2.3. Упрощение произведений матриц Дирака и спиноров

<i>DiracOrder</i> {expr}	$\rightarrow$ упорядочивание матриц Дирака по алфавиту;
<i>DiracOrder</i> {expr,a,b,...}	$\rightarrow$ упорядочивание по порядку a,b,...;
<i>DiracSimplify</i> {expr}	$\rightarrow$ свертка всех парных лоренцевских индексов и упрощение.

$\gamma_\beta \gamma_\alpha \rightarrow 2g_{\alpha\beta} - \gamma_\alpha \gamma_\beta$	In [1]:=DiracOrder [DiracMatrix [be, al]]
	Out [1]=2g [al, be]-ga [al] ga [be]
	In [2]:=DiracOrder [%, {be, al}]
	Out [2]=ga [be] ga [al]
$\gamma^\mu \gamma_\mu \hat{p} \hat{p} = 4p^2$	In [3]:=DiracOrder [DiracMatrix [mu, nu], DiracSlash [p, p]]
	Out [3]=4p . p
$\gamma^\alpha \gamma_\mu \gamma_\alpha = -2\gamma_\mu$	In [4]:=DiracOrder [DiracMatrix [al, mu, al]]
	Out [4]=-2ga [mu]
$-\hat{p} \hat{q} \hat{p} = \hat{q} p^2 - 2\hat{p} (p \cdot q)$	In [5]:=DiracOrder [DiracSlash [-p, q, p]]
	Out [5]=gs [q] p . p-2gs [p] p . q

При вычислениях выражений, включающих  $\gamma_5$ , используется функция *DiracSimplify*. Все матрицы  $\gamma_5, \gamma_6, \gamma_7$  передвигаются в правый конец произведения матриц Дирака. При упрощении выражений функция *DiracSimplify* учитывает уравнение Дирака  $(\hat{p} - m)u(p, m) = 0$ .

$\gamma_\mu \gamma^\mu = 4$	In [1]:=DiracSimplify [DiracMatrix [mu, mu]]
	Out [1]=4
$\gamma^\mu \gamma_\nu \gamma_\rho \gamma_\sigma \gamma_\mu = -2\gamma_\sigma \gamma_\rho \gamma_\nu$	In [2]:=DiracSimplify [DiracMatrix [mu, nu, ro, si, mu]]

$\gamma_5 \gamma_\mu = -\gamma_\mu \gamma_5$	Out [2] = -2ga[si]ga[ro]ga[nu] In [3] := DiracSimplify[ DiracMatrix[5], DiracMatrix[mu]]
$\gamma_6 \gamma_\nu \gamma_\tau \gamma_\mu = \gamma_\nu \gamma_\mu \gamma_6$	Out [3] = -ga[mu]ga[5] In [4] := DiracSimplify[ DiracMatrix[6, nu, 7, mu]]
$(\hat{p} - m)u(p, m) = 0$	Out [4] = ga[nu]ga[mu]ga[6] In [5] := DiracSimplify[ (DiracSlash[p] - m).Spinor[p, m]]
$\bar{v}(p, m)\hat{q}\hat{p}u(q, M)$ $\bar{v}(p, m)u(q, M)$ $[2(p.q) + mM]$	Out [5] = 0 In [6] := DiracSimplify[ Spinor[-p, m].DiracSlash[q, p]. Spinor[q, M]]//Factor Out [6] = v[p, m]u[q, M] (mM + 2p.q)

### 23.2.4. Вычисление следа произведения матриц Дирака

Основные функции:

*DiracTrace*[expr] → ввод следа произведения матриц Дирака;  
*Tr*[expr] → вычисление следа произведения матриц Дирака.

Функция *DiracTrace* непосредственно не вычисляет след выражения expr. Напрямую это можно сделать, используя функцию *Tr*. Команда *EvaluateDiracTrace* вычисляет след выражения, определенного функцией *DiracTrace*.

$tr(\gamma^\alpha \gamma^\beta) = 4g^{\alpha\beta}$	In [1] := Tr [DiracMatrix[a1, be]] Out [1] = 4g [a1, be]
$tr(\hat{a}\hat{b}\hat{c}\hat{d}) =$ $4[(a.b)(c.d) - (a.c)(b.d)$ $+ (a.b)(c.d)]$	In [2] := Tr [DiracSlash[a, b, c, d]] Out [2] = 4 a.d b.c - 4 a.c b.d + 4 a.b c.d
$tr(\gamma^\mu \gamma^\nu \gamma^\rho \gamma^\sigma \gamma^5) =$ $= -4i\varepsilon^{\mu\nu\rho\sigma}$	In [3] := Tr [DiracMatrix[mu, nu, ro, si, 5]] Out [3] = -4Ieps [mu, nu, ro, si]

Пример вычисления более сложного выражения:

$$T = tr[(\hat{q} + m)\gamma^\alpha(\hat{p} + \hat{k} + m)\gamma^\beta(\hat{p} + m)\gamma^\beta(\hat{p} + \hat{k} + m)\gamma^\alpha].$$

Для удобства введем новые переменные

```
In[1]:= (Q=DiracSlash[q];
P=DiracSlash[P]; K=DiracSlash[k];
A=DiracMatrix[al]; B=DiracMatrix[be] )
Out[1]=ga[be]
```

Ввод T

```
In[2]:= T=Tr[Q+m, A, P+K+m, B, P+m, B,
P+K+m, A]
```

Получаем

```
Out[2]=4m^4+4m^2k.k+4m^2k.k+4m^2k.p
-4m^2 k.q+ 2k.p k.q+2k.q p.q-3m^2p.q-
k.k p.q+p.p q.q
```

Если след  $tr(\gamma^\alpha \gamma^\beta \gamma^\rho \gamma^\sigma)$  не вычисляется.

```
In[3]:= DiracTrace[
DiracMatrix[al, be, ro, si]]
Out[3]=tr[ga[al]ga[be]ga[ro]ga[si]]
```

Свернем след с  $g^{\alpha\beta}$

```
In[4]:= Contract[%
MetricTensor[al, be]]
Out[4]=tr[ga[be]ga[be]ga[ro]ga[si]]
```

Теперь вычислим след

```
In[5]:= EvaluateDiracTrace[%]
Out[5]=16g[ro, si]
```

Для описания процессов рассеяния  $2 \rightarrow 2$  часто используют набор переменных Мандельштама  $s, t, u$ :

*SetMandelstam* $\{s, t, u, p_1, p_2, p_3, p_4, m_1, m_2, m_3, m_4\}$   $\rightarrow$  определяет скалярные произведения 4-векторов через  $s, t, u$  и полагает  $p_i^2 = m_i^2$ ;

*TrickMandelstam* $[expr, \{s, t, u, m1^2 + m2^2 + m3^2 + m4^2\}]$   $\rightarrow$  приводит выражение  $expr$  к наиболее компактному виду, используя формулу  $s + t + u = m_1^2 + m_2^2 + m_3^2 + m_4^2$ .

Предполагается, что все 4-импульсы "входящие", т.е.  $p_1 + p_2 + p_3 + p_4 = 0$  и  $s = (p_1 + p_2)^2$ ,  $t = (p_1 + p_3)^2$ ,  $u = (p_1 + p_4)^2$ .

Вычислим выражение  $W = tr(\hat{p}_1 \hat{p}_2 \hat{p}_3 \hat{p}_4)$  для процесса  $12 \rightarrow 34$ , где  $p_1^2 = p_2^2 = m^2$ ,  $p_3^2 = p_4^2 = M^2$ , используя *SetMandelstam*:

Определение  $s, t, u$   $In[6] := SetMandelstam[s, t, u, p_1, p_2, -p_3, -p_4, m, m, M, M]$

```
Out[6]={{m^2, -m^2 + s, m^2 + m^2 - t, m^2 + m^2 - u,
m^2 + m^2 - u, m^2 + m^2 - t, -m^2 + s, m^2}}
```

Ввод W	In[7]:= W=Tr[DiracSlash[p1,p2,p3,p4]] Out[7]=4m <sup>2</sup> M <sup>2</sup> -2m <sup>2</sup> s-2M <sup>2</sup> s+s <sup>2</sup> +2m <sup>2</sup> t+ 2M <sup>2</sup> t-t <sup>2</sup> -2m <sup>2</sup> u-2M <sup>2</sup> u+u <sup>2</sup>
Исключим u	In[8]:= W1=W/.u ->-t-s+2m <sup>2</sup> +2M <sup>2</sup> Out[8]=4m <sup>2</sup> M <sup>2</sup> -4m <sup>2</sup> s-4M <sup>2</sup> s+2s <sup>2</sup> +2st
Упрощение	In[9]:=TrickMandelstam[(s+t-u)(2m-t-u), {s,t,u,2m^2}] Out[9]=2s(m-u)
Результат всегда факторизуется	In[10]:=TrickMandelstam[m^2s- s^2+m^2t- st+m^2u-su, {s,t,u,2m^2 } Out[10]=2m <sup>2</sup> (m <sup>2</sup> -s)

### 23.2.5. Вычисление следа матриц Гелл-Манна

Основные функции:

*GellMannTrace [expr]* → вычисление следа произведения матриц Гелл-Манна.

$tr(\lambda_a \lambda_b) = 2\delta_{ab}$	In[1]:=GellMannTrace[ GellMannMatrix[a,b]] Out[1]= 2d[a,b]
$tr(\lambda_a \lambda_b \lambda_a \lambda_b) = -\frac{32}{3}$	In[2]:= GellMannTrace[ GellMannMatrix[a,b,a,b]] Out[2]= $-\frac{32}{3}$
$tr(\lambda_a \lambda_b \lambda_c f_{abc}) = 48i$	In[3]:=GellMannTrace[SU3F[a,b,c] GellMannMatrix[a,b,c]] Out[3]=48I
$tr(f_{acd} f_{bcd}) = 3f_{acd} f_{bcd} = 9\delta_{ab}$	In[4]:=GellMannTrace[ SU3F[a,c,d]SU3F[b,c,d]] Out[4]=9d[a,b]
$f_{acd} f_{bcd} = 3\delta_{ab}$	In[4]:=SU3F[a,c,d]SU3F[b,d,c] Out[4]=-(tr[la[a]la[c]la[d]]+ tr[la[a]la[d]la[c]]) (-tr[la[b]la[d]la[c]]+ tr[la[b]la[c]la[d]])



Определим  $f$  как SU3F

$$f_{abr}f_{rcs} + f_{bcr}f_{ras} + f_{car}f_{rbs} = 0$$

$$\lambda_a \lambda_b \lambda_a = -\frac{2}{3} \lambda_b$$

$$\text{tr}(\lambda_c \lambda_b \lambda_c \lambda_a \lambda_b \lambda_a)$$

In[5]:=ExpandAll[%]

Out[5]=3d[a,b]

In[6]:=f[x\_]:=SU3F[x]

In[7]:=ExpandAll[

f[a,b,r]f[r,c,s]+f[b,c,r]f[r,a,s]+

f[c,a,r]f[r,b,s]]

f[c,a,r]f[r,b,s]]

Out[7]=0

In[8]:=GellMannMatrix[a,b,a]

Out[8]= $-\frac{21a[b]}{3}$

In[9]:=GellMannTrace[

GellMannMatrix[c,b,c,a,b,a]]

Out[9]= $\frac{64}{9}$

### 23.3. Отличия FeynCalc 3.0 от FeynCalc 1.0

В версии пакета FeynCalc 3.0, работающей под управлением системы Mathematica 3.0, в отличие от версии FeynCalc 1.0, работающей под управлением системы Mathematica 2.2, упрощен ввод основных функций путем сокращения заголовков:

<i>FourVector</i> [p,mu]	→ <i>FV</i> [p,mu];
<i>LeviCivita</i> [mu,nu,ro,si]	→ <i>LC</i> [mu,nu,ro,si];
<i>MetricTensor</i> [mu,nu]	→ <i>MT</i> [mu,nu];
<i>ScalarProduct</i> [p,q]	→ <i>SP</i> [p,q];
<i>DiracMatrix</i> [mu,nu,...]	→ <i>GA</i> [mu,nu,...];
<i>DiracSlash</i> [p,q,...]	→ <i>GS</i> [p,q ...];
<i>DiracMatrix</i> [5]	→ <i>GA</i> [5];
<i>DiracMatrix</i> [6]	→ <i>GA</i> [6];
<i>DiracMatrix</i> [7]	→ <i>GA</i> [7].

В версии FeynCalc 3.0 появился ряд новых функций, в частности:

*GluonGhostVertex* [{p,mu,a},{q,nu,b},{k,la,c}] → дух-глюонная  
или *GGV* [{p,mu,a},{q,nu,b},{k,la,c}] вершина в КХД  
 $g_s k^\mu f_{abc}$

**QuarkGluonVertex**[ $\mu, a$ ] → кварк-глюонная  
 или **QGV**[ $\mu, a$ ] вершина в КХД  $ig_s T_a \gamma^\mu$ ;  
**GluonVertex** [{ $p, \mu, a$ },{ $q, \nu, b$ },{ $k, \lambda, c$ }] → трехглюонная  
 или **GV** [{ $p, \mu, a$ },{ $q, \nu, b$ },{ $k, \lambda, c$ }] вершина в КХД  
 $g_s f_{abc} [(p - q)^\lambda g^{\mu\nu} +$   
 $(q - k)^\mu g^{\lambda\nu} + (k - p)^\nu g^{\lambda\mu}]$ ;

**FermionSpinSum**[*expr*] → суммирование по поляризациям фермионов в выражении *expr*, которое является квадратом модуля амплитуды.

В версиях **FeynCalc** старше 1.0 изменен синтаксис определений и команд для операций с матрицами-генераторами фундаментального представления группы  $SU(N)$ . Матрицы Гелл-Манна отвечают группе  $SU_c(3)$ , это задается параметром  $SUNN=N=3$ .

Основные функции:

**SUNN** →  $N$ , число цветов группы  $SU(N)$ ;  
**CA** →  $C_A=N$ ;  
**CF** →  $C_F=(N^2-1)/2N$ ;  
**SUNT**[*a*] →  $T_a$ , генератор в фундаментальном представлении группы  $SU(N)$ ;  
**SUND**[*a,b,c*] →  $d_{abc}$ , структурная константа группы  $SU(N)$ ;  
**SUNDelta**[*a,b*] → символ Кронеккера  $\delta_{a,b}$ ;  
**SUNDeltaContract**[*expr*] → свертка по индексам  $SU(N)$ ;  
**SUNF**[*a,b,c*] →  $f_{abc}$ , структурная константа группы  $SU(N)$ ;  
**SUNTrace**[*expr*] → определение следа цветowych матриц  $SU(N)$ , по умолчанию след не вычисляется;  
**SUNIndex**[*a*] → объявление  $SU(N)$  индекса;  
**SUNSimplify** → упрощение матричного выражения, свертка по парным цветовым индексам, вычисление следа, если **SUNTrace** → **True**.

Некоторые примеры:

```
<< D:\WNMATH30\HighEnergyPhysics\fc.m
```

```
In[1]:= SUNTrace[SUNT[a, b]]
```

```
Out[1]= $\frac{\delta_{ab}}{2}$ 
```

```
In[2]:= SUNSimplify[SUNT[a, b, a, b]]
```

```
Out[2]= $-\frac{1}{2}(C_A - 2C_F)C_F$ 
```

```
In[3]:= SUNSimplify[SUNT[a, b, a, b], SUNNTOCACF -> False]
```

```
Out[3]= $-\frac{N^2-1}{4N^2}$ 
```

```
In[4]:= t = SUNTrace[SUNT[a,b,c],Explicit -> True]SUNF[a,b,c]
```

```
Out[4]= $f_{abc} \left( \frac{1}{4}d_{abc} + \frac{1}{4}i f_{abc} \right)$ 
```

```
In[5]:= SUNSimplify[t, Explicit -> True]
```

```
Out[5]= $\frac{1}{2}iC_A^2C_F$ 
```

```
In[6]:= SUNSimplify[SUNF[a, c, d] SUNF[b, c, d]]
```

```
Out[6]= $C_A\delta_{ab}$ 
```

```
In[7]:= SUNT[a, b, a] // SUNSimplify
```

```
Out[7]= $-\frac{1}{2}(C_A - 2C_F)T_b$ 
```

```
In[8]:= SUNTrace[SUNT[a, b, c], Explicit -> True]
```

```
Out[8]= $\frac{1}{4}d_{abc} + \frac{1}{4}if_{abc}$ 
```

```
In[9]:= SUNTrace[SUNT[a, b, c], Explicit -> False]
```

```
Out[9]= $tr(T_a \cdot T_b \cdot T_c)$ 
```

### III. Примеры решения задач

#### 24. Обработка экспериментальных данных

##### Задание 1

Провести статистический анализ распределения числа частиц, регистрируемых счетчиком при измерении естественного радиоактивного фона Земли.

Проводятся четыре серии по сто измерений срабатываний счетчика с интервалами однократного измерения 5, 10, 15 и 20 с соответственно (всего 400 измерений).

Для каждой серии измерений рассчитываются:

1. среднее количество отсчетов

$$\langle n \rangle = \frac{1}{N} \sum_{i=1}^N n_i,$$

где  $N$  — количество измерений в одной серии;

2. дисперсия

$$D = \sigma^2 = \frac{1}{N-1} \sum_{i=1}^N (n_i - \langle n \rangle)^2;$$

3. стандартное отклонение  $\sigma = \sqrt{D}$ ;

4. массив  $P(n_i)$  значений распределения Пуассона

$$P(n_i) = \frac{\langle n \rangle^{n_i}}{n_i!} e^{-\langle n \rangle}.$$

На основании экспериментальных данных строятся четыре гистограммы.

Для этого диапазон имеющихся значений  $n_i$  в каждой серии (от 0 до максимального) разбивают на некоторое произвольное число равных интервалов группировки  $\Delta n_i$ . Затем последовательно от первого до последнего перебирают значения случайной величины и смотрят,

в какой интервал попадает каждое значение. Число значений  $n_i$ , принадлежащих данному интервалу  $\Delta n_i$ , называется групповой частотой.

Если теперь по оси абсцисс обозначить интервалы  $\Delta n_i$ , и над каждым из них нарисовать прямоугольник, высота которого равна соответствующей групповой частоте, т.е. числу случаев попадания значений случайной величины  $n_i$  в данный интервал, то полученная система прямоугольников образует гистограмму. Подгружаем дополнительные пакеты системы Mathematica:

```
In[1] := <<... \wnmath22\packages\statisti\master.m
In[2] := <<... \wnmath22\packages\graphics\master.m
```

Вводим с клавиатуры экспериментальные данные первой серии:

```
In[3] := ser1={6, 5, 6, 2, 3, 7, 3, 1, 5, 6, 3, 2, 3, 3,
  3, 2, 5, 5, 1, 5, 8, 3, 3, 3, 8, 5, 4, 2, 4, 5,
  2, 2, 6, 8, 6, 9, 4, 4, 3, 6, 6, 6, 3, 2, 7, 4,
  0, 5, 7, 5, 4, 4, 3, 0, 3, 3, 4, 1, 4, 3, 7, 4,
  2, 4, 1, 5, 5, 4, 2, 2, 5, 5, 3, 4, 3, 2, 7, 4,
  8, 6, 3, 3, 3, 3, 4, 7, 3, 5, 3, 3, 4, 3, 2, 1,
  5, 4, 7, 4, 4, 3};
```

Находим среднее количество отсчетов:

```
In[4] := nsr1=N[Mean[ser1]]
Out[4] = 4.02,
```

а также дисперсию:

```
In[5] := d=N[Variance[ser1]]
Out[5] = 3.57535
```

стандартное отклонение:

```
In[6] := sigma=N[StandardDeviation[ser1]]
Out[6] = 1.89086
```

и массив  $P(n_i)$  значений распределения Пуассона:

```
In[7] := For[i=1,i<101,i++,{p=N[PDF[PoissonDistribution[nsr1],
  ser1[[i]]]],Print[ser1[[i]]," ",i," ",p]]}
```

```

Out[7] = 6 1 0.105235
         5 2 0.157067
         6 3 0.105235
         2 4 0.145064
         3 5 0.194385
         7 6 0.0604349
         3 7 0.194385
         1 8 0.0721709
         . . .
         . . .
         . . .
         3 100 0.194385

```

Найдем групповые частоты для каждого интервала группировки  $\Delta n_i$ .

Пусть величина каждого интервала группировки  $\Delta n_i = 1$ :

```

In[8] := dn=1;
In[9] := gis1= BinCounts[ser1,\{0,Max[ser1],dn}]
Out[9] = {5, 12, 26, 19, 15, 9, 7, 4, 1}

```

Для определения границ интервалов группировки при заданном  $\Delta n_i$  используем команду:

```

In[10] := CumulativeSums[Table[1,{Length[gis1]}]]

```

(здесь первый интервал начинается с нуля):

```

Out[10] = {1, 2, 3, 4, 5, 6, 7, 8, 9}

```

По полученным интервалам группировки строим гистограмму:

```

In[11] := BarChart[gis1,PlotLabel->"<n>=4.02"]

```

Аналогичные расчеты проводим для экспериментальных данных других серий.

Гистограммы для всех четырех серий измерений приведены на рис. 1.

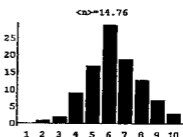
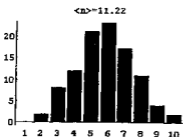
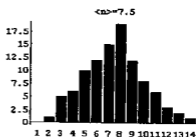
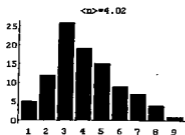


Рис. 1

Все экспериментальные данные можно свести в табл. 8.

Таблица 8

Номер серии: 1		Время одного наблюдения: 5 с		Количество наблюдений: 100		
N	Интервал группировки	Групповая частота	Распред. Пуассона	Среднее	D	$\sigma$
1	(0, 1]	5	0.072	4.02	3.58	1.89
2	(1, 2]	12	0.145			
3	(2, 3]	26	0.194			
⋮	⋮	⋮	⋮			

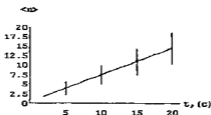
### Задание 2

Проверить линейную зависимость величины  $\langle n \rangle$  от времени наблюдения. Начертить график  $\langle n \rangle(t)$ , где  $\langle n \rangle$  — среднее число частиц в каждой серии опытов,  $t$  — время наблюдения в соответствующей серии.

```
In[12] := p:=ErrorListPlot[{{5,4.02,1.88},{10,7.5,2.5},
{15,11.22,3.46},{20,14.76,4.15}},PlotRange->
{{0,21},{0,20}}];
```

```
f=Fit[{{5,4.02},{10,7.5},{15,11.22},{20,14.76}},
{1,t},t]
g:=Plot[f,{t,2,20},PlotRange->{{0,20},{0,20}}];
Show[{p,g},AxesLabel->{"t,(c)","<n>}]
```

Out[12] = 0.39 + 0.7188 t



Out[13] = - Graphics -

### Задание 3

Определить экспериментально среднее время между регистрациями частиц и функцию распределения интервалов времени между двумя соседними регистрациями.

Проводятся не менее ста измерений интервалов времени  $t_i$  между соседними импульсами счетчика.

Строится экспериментальная вероятностная гистограмма. Для этого весь диапазон значений  $t_i$  времен молчаний счетчика (от 0 до  $t_{i,max}$ ) следует разбить на одинаковые отрезки  $\Delta t_i$  — интервалы группировки.

Подсчет групповых частот, т.е. событий, соответствующих каждому интервалу  $\Delta t_i$ , производится следующим образом: случайным событием в нашей схеме является отсутствие срабатывания счетчика в моменты времени, предшествующие регистрации частицы. Следовательно, если счетчик зарегистрировал частицу в момент времени  $t_i \in \Delta t_i$ , то в этом эксперименте мы наблюдали событие (отсутствие импульса) в каждом интервале  $\Delta t_1, \Delta t_2, \dots, \Delta t_{i-1}$ . Число значений  $t_i$ , принадлежащих данному интервалу  $\Delta t_i$ , называется групповой частотой.

Построим экспериментальную вероятностную гистограмму (на горизонтальной оси откладываем интервалы группировки) и сравним ее с теоретической кривой.

Вводим экспериментальные данные:



```
In[14] := time={0.121, 0.069, 0.244, 0.499, 0.440, 0.146,
0.043, 0.198, 0.596, 0.584, 0.058, 0.149, 0.041,
0.117, 0.045, 0.101, 0.004, 0.010, 0.519, 0.418,
0.290, 0.281, 0.713, 0.534, 0.090, 0.142, 0.096,
0.004, 0.161, 1.195, 0.044, 0.159, 0.209, 0.023,
0.775, 0.005, 0.134, 0.312, 0.390, 0.290, 0.125,
0.279, 0.616, 0.519, 0.307, 1.518, 0.900, 1.218,
0.130, 0.207, 0.264, 0.912, 0.666, 0.211, 0.168,
0.174, 0.112, 0.165, 0.191, 0.164, 0.739, 0.366,
0.342, 0.396, 0.200, 0.025, 0.443, 0.633, 0.464,
0.025, 0.487, 0.658, 0.144, 0.191, 0.057, 0.771,
0.177, 0.829, 0.312, 0.149, 0.039, 0.163, 0.049,
0.867, 0.054, 0.310, 0.141, 0.047, 0.258, 0.238,
0.230, 0.072, 0.402, 0.181, 0.341, 0.098, 0.112,
0.992, 0.194, 0.115};
```

Вводим величину интервала группировки  $\Delta t$  при числе интервалов 10:

```
In[15] := dn=Max[time]/10;
```

Подсчитываем интервалы группировки, учитывая, что случайное событие — отсутствие импульса:

```
In[16] := gis=BinCounts[time,{0,Max[time],dn}]
Out[16] = {37, 26, 13, 8, 6, 5, 2, 1, 1, 1}
```

Вычислим по данным эксперимента среднее время между двумя ближайшими импульсами  $\langle t \rangle$ , среднее число частиц в единицу времени  $n_1 = \frac{1}{\langle t \rangle}$ , статистические отклонения. Среднее время между двумя ближайшими импульсами  $\langle t \rangle$ :

```
In[17] := tsr=N[Mean[time]]
Out[17] = 0.314498
```

Среднее число частиц в единицу времени:

```
In[18] := n1=1/tsr
Out[18] = 3.17967
```

Статистические отклонения:

```
In[19] := d=N[Variance[time]]
Out[19] = 0.0894447
In[20] := sigma=StandardDeviation[time]
Out[20] = 0.299073
```

Вычислим теоретические значения статистических отклонений по формулам, приведенным в задании, и сравним полученные значения с экспериментальными статистическими отклонениями.

```
In[21] := dteor=1/n1^2
Out[21] = 0.0989089
In[22] := sigmateor=N[Sqrt[dteor]]
Out[22] = 0.314498
```

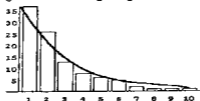
Построим для найденного значения  $n_1$  теоретическую кривую экспоненциального распределения и сравним ее с экспериментальной гистограммой.

Для построения сгенерируем экспоненциальное распределение случайных чисел (их количество определяется числом измерений) и аппроксимируем точки гистограммы полиномом 3-й степени:

```
In[23] := teor=Table[Random[ExponentialDistribution[n1]],{100}];
          dn=Max[teor]/10;
          gisteor=BinCounts[teor,{0,Max[teor],dn}];
          f=Fit[gisteor,{1,x,x^2,x^3},x]
Out[23] = 43.5 - 13.966 x + 1.67599 x2 - 0.0701243 x3
```

Выведем на экран экспериментальную гистограмму и соответствующую ей теоретическую кривую:

```
In[24] := Show[BarChart[gis,BarStyle->{RGBColor[1,1,1]}],
               Plot[f,{x,0,10},PlotStyle->Thickness[0.007]],
               PlotRange->{0,Max[{gis,gisteor}}]]
```



```
Out[24] = - Graphics -
```

#### Задание 4

Изучить форму графиков различных распределений.

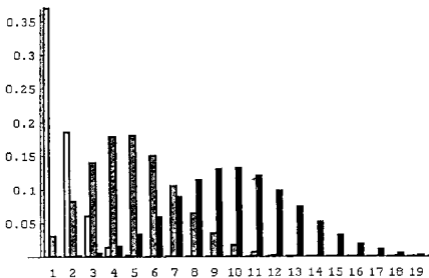
Строим графики для нескольких значений параметров распределений.

##### 1. Биномиальное распределение

1)  $N = \text{const}$ ,  $p$  меняется ( $0 < p < 1$ )

```
In[25] := data1=Table[PDF[BinomialDistribution[100,0.01],i],
  {i,20}];
  data2=Table[PDF[BinomialDistribution[100,0.05],i],
  {i,20}];
  data3=Table[PDF[BinomialDistribution[100,0.1],i],
  {i,20}];
  BarChart[data1,data2,data3,PlotLabel->"N=100",
  BarStyle->{RGBColor [1,1,1],RGBColor [0.5,0.5,0.5],
  RGBColor [0,0,0]},
  PlotRange->{{0,20},{0,Max[data1,data2,data3]}}];
```

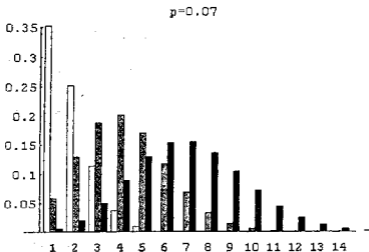
N=100



Out[25] = - Graphics -

2)  $p = \text{const}$ ,  $N$  меняется

```
In[26] := data1=Table[PDF[BinomialDistribution[20,0.07],i],  
  {i,15}];  
  data2=Table[PDF[BinomialDistribution[60,0.07],i],  
  {i,15}];  
  data3=Table[PDF[BinomialDistribution[100,0.07],i],  
  {i,15}];  
  BarChart[data1,data2,data3,PlotLabel->"p=0.07",  
  BarStyle->{RGBColor [1,1,1],RGBColor [0.5,0.5,0.5],  
  RGBColor [0,0,0]},  
  PlotRange->{{0,15},{0,Max[data1,data2,data3]}}];
```

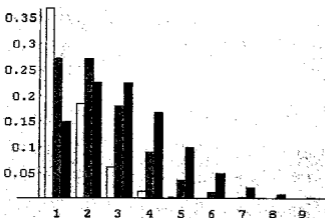


Out[26] = - Graphics -

2. Распределение Пуассона и Лапласа для разных  $\langle n \rangle$

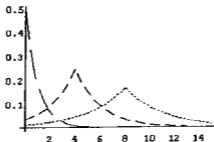
```
In[27] := data1=Table[PDF[PoissonDistribution[1],i],  
  {i,10}];  
  data2=Table[PDF[PoissonDistribution[2],i],{i,10}];  
  data3=Table[PDF[PoissonDistribution[3],i],{i,10}];  
  BarChart[data1,data2,data3,BarStyle->
```

```
{RGBColor [1,1,1],RGBColor [0.5,0.5,0.5],
RGBColor [0,0,0]},PlotRange->{{0,10},{0,Max[data1,
{0,Max[data1,data2,data3]}}];
```



Out[27] = - Graphics -

```
In[28] := f1=PDF[LaplaceDistribution[0,1],x];
f2=PDF[LaplaceDistribution[4,2],x];
f3=PDF[LaplaceDistribution[8,3],x];
Plot[{f1,f2,f3},{x,0,15},PlotStyle->
{Dashing[{0.15,0.05}],Dashing[{0.05,0.03}],
Dashing[{0.01,0.01}]}
```

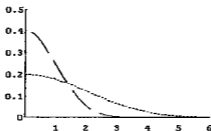


Out[28] = - Graphics -

### 3. Нормальное распределение.

1) для двух независимых величин  $x, y$ , нормально распределенных с параметрами  $(0, \sigma_x^2)$  и  $(0, \sigma_y^2)$ , если  $\sigma_x^2 \geq \sigma_y^2$ .

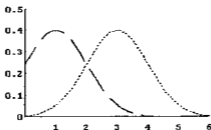
```
In[29] := f1=PDF[NormalDistribution[0,1],x];  
          f2=PDF[NormalDistribution[0,2],x];  
          Plot[{f1,f2},{x,0,6},PlotRange->{{0,6},{0,0.5}},  
              PlotStyle->{Dashing[{0.15,0.05}],  
                           Dashing[{0.01,0.01}]}
```



Out[29] = - Graphics -

2) для двух значений  $a_1 < a_2$  при  $\sigma_1 = \sigma_2$ .

```
In[30] := f1=PDF[NormalDistribution[1,1],x];  
          f2=PDF[NormalDistribution[3,1],x];  
          Plot[{f1,f2},{x,0,6},PlotRange->{{0,6},{0,0.5}},  
              PlotStyle->{Dashing[{0.15,0.05}],  
                           Dashing[{0.01,0.01}]}
```



Out[30] = - Graphics -

## 25. Задачи по общей физике

### 25.1. Движение спутника Земли

Рассмотрим задачу о движении точечного тела массы  $m$  (спутника) в поле тяготения бесконечно тяжелой и неподвижной Земли, т.е.  $m \ll M$ , где  $M$  — масса Земли. В полярных координатах уравнения движения имеют вид:

$$\begin{aligned}\ddot{r} - r\dot{\phi}^2 &= -\frac{\gamma M}{r^2}, \\ 2\dot{r}\dot{\phi} + r\ddot{\phi} &= 0,\end{aligned}\quad (1)$$

а в декартовых:

$$\begin{aligned}\ddot{x} &= -\frac{\gamma M x}{(x^2 + y^2)^{3/2}}, \\ \ddot{y} &= -\frac{\gamma M y}{(x^2 + y^2)^{3/2}}.\end{aligned}\quad (2)$$

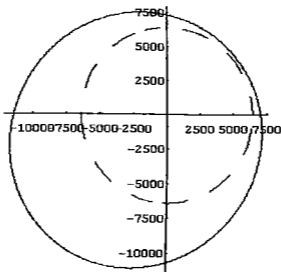
(3)

Произведение постоянной всемирного тяготения  $\gamma$  на массу Земли  $M$  выражается через ускорение свободного падения  $g$  у поверхности Земли:  $\gamma M = gR^2 = 6.3 \cdot 10^7 (\text{м}^3/\text{сек}^2) = 226 (\text{км}^3/\text{мин}^2)$ .

Для решения уравнений (1) и (2) необходимо задать начальные условия. Пусть при  $t=0$  спутник находится на расстоянии  $r(0) = 7000$  км от центра Земли в точке с декартовыми координатами  $(x, y) = (7000, 0)$  и имеет скорость  $\vec{V}(0) = (\dot{x}(0), \dot{y}(0))$ , где  $\dot{x}(0) = -0.03$  (км/мин) и  $\dot{y}(0) = 0.15$  (км/мин). Найдем траекторию движения спутника и зависимость расстояния от спутника до поверхности Земли ( $R = 6400$  км), решая уравнения движения в декартовых координатах:

```
In[1] := eq1=x''[t]==-226 x[t]/Sqrt[x[t]^2+y[t]^2]^3;
In[2] := eq2=y''[t]==-226 y[t]/Sqrt[x[t]^2+y[t]^2]^3;
In[3] := f=NDSolve[{eq1,eq2,x[0]==7000,y[0]==0,x'[0]==-0.03,
  y'[0]==0.20},{x,y},{t,0,5 10^5},MaxSteps->5000];
In[4] := s=ParametricPlot[Evaluate[{x[t],y[t]}/.f],{t,0,5 10^5},
  PlotPoints->1000];
In[5] := z=ParametricPlot[Evaluate[{6400 Cos[t],6400 Sin[t]}],
```

```
{t,0,2 Pi},PlotStyle->Dashing[{0.1,0.09}]]];
In[6] := Show[{s,z},AspectRatio->Automatic]
```



Out[6]= - Graphics -

## 25.2. Электростатическое поле системы зарядов

**Задача 1** Заряд распределен в области  $r < R$  по закону  $\rho(r) = c \exp(-(R-r)/a) \cos(\theta)$ . Найти дипольный и квадрупольный моменты системы.

Известно, что потенциал системы зарядов на большом расстоянии от нее ( $r \gg R$ ) с точностью до членов порядка  $(R/r)^3$  записывается в виде:

$$\Phi(\vec{r}) = \int \frac{\rho(\vec{\xi}) d^3\xi}{|\vec{r} - \vec{\xi}|} \approx \frac{q}{r} + \frac{(\vec{p}\vec{r})}{r^3} + \frac{1}{2r^5} Q_{ij} r^i r^j, \quad (4)$$

где  $q$  — полный заряд системы,  $\vec{p}$  — дипольный момент системы,  $Q_{ij}$  — компоненты квадрупольного момента системы ( $i, j = 1, 2, 3$  и  $r^1 = x, r^2 = y, r^3 = z$ ). Разложим знаменатель подынтегрального выражения в ряд по  $1/r$  до членов  $1/r^3$ , а затем проинтегрируем по  $\vec{\xi}$  в пределах объема, где находится заряд:



```

In[1] := pl=c*Exp[-(R-r1)/a]*Cos[t1]
          c Cos[t1]
Out[1] = -----
          (R - r1)/a
          E
In[2] := d=PowerExpand[Normal[Series[1/Sqrt[r^2+r1^2-2 r r1 Z],
          {r1,0,3}]]];
In[3] := d=Collect[Simplify[d],r]
          2      2 2      3      3 3
          -r1   3 r1 Z   -3 r1 Z   5 r1 Z
          ---- + ----- + ----- + -----
          2      2      2      2
Out[3] = - + ---- + ----- + -----
          r   2      3      4
          r      r      r
In[4] := Z=Sin[t]*Sin[t1]*(Cos[f]*Cos[f1]+Sin[f]*Sin[f1])
          +Cos[t]*Cos[t1];
In[5] := w=Integrate[pl*d*r1^2*Sin[t1],{r1,0,R},{t1,0,Pi},
          {f1,0,2 Pi}]
Out[5] =
          4      3      2      2 3
          8 a c Pi Cos[t]  4 a c Pi (-6 a + 6 a R - 3 a R + R ) Cos[t]
          ----- + -----
          R/a 2      2
          E   r      3 r
In[6] := p=Collect[Simplify[w*r^2/Cos[t],{Pi,c}]
          4      3
          4 8 a      3      2 2 4 a R
Out[6] = c Pi (-8 a + ---- + 8 a R - 4 a R + -----)
          R/a      3
          E

```

В полученной формуле для потенциала поля системы зарядов в точке со сферическими координатами  $r, \theta, \phi$  присутствует только член разложения  $\sim 1/r^2$ , следовательно  $q = 0$ ,  $Q_{ij} = 0$  и дипольный

момент имеет только z-проекцию ( $\sim c \dots$ )

$$p = c\pi \left( -8a^4 + 8a^4 e^{-R/a} + 8a^3 R - 4a^2 R^2 + \frac{4aR^3}{3} \right). \quad (5)$$

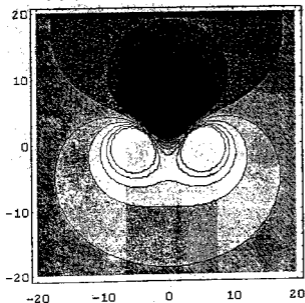
**Задача 2** Построить контурный график распределения потенциала в плоскости  $z = 0$  для системы трех зарядов  $q_1 = Q$ ,  $q_2 = Q$ ,  $q_3 = -2Q$  с координатами  $\vec{r}_1 = (5, 0)$ ,  $\vec{r}_2 = (-5, 0)$ ,  $\vec{r}_3 = (0, 5)$ .

Потенциал в точке на плоскости определяется как сумма потенциалов, создаваемых каждым из трех точечных зарядов:

$$\Phi(x, y, z) = \sum_{i=1}^3 \frac{q_i}{\sqrt{(x-x_i)^2 + (y-y_i)^2}}. \quad (6)$$

In[1] := x1={5,-5,0};y1={0,0,5};q={1,1,-2};

In[2] := ContourPlot[Sum[q[[i]]/Sqrt[(x1[[i]]-x)^2+(y1[[i]]-y)^2],  
{i,1,3}],{x,-10,10},{y,-10,10},PlotPoints->100]



Out[2] = - ContourGraphics -

**Задача 3** Построить контурный график распределения потенциала на плоскости для бесконечно тонкого заряженного эллипса  $x^2/a^2 + y^2/b^2 = 1$ , где  $a/b = 5$ .

Расчетная формула получается из общего выражения:

$$\Phi(\vec{r}) = \int \frac{\lambda dl(\vec{\xi})}{|\vec{r} - \vec{\xi}|} \quad (7)$$

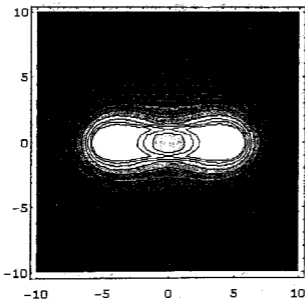
переходом к интегрированию по полярному углу  $\phi$

$$\Phi(x, y) = \lambda \int_0^{2\pi} \frac{\sqrt{a^2 \cos^2(\phi) + b^2 \sin^2(\phi)} d\phi}{|\vec{r} - \vec{\xi}|}, \quad (8)$$

где  $\vec{r} = (x, y)$  и  $\vec{\xi} = (a \cos(\phi), b \sin(\phi))$ ,  $\lambda$  — линейная плотность заряда.

```
In[1] := x1=5 Cos[f];y1=1 Sin[f];
```

```
In[2] := ContourPlot[NIntegrate[Sqrt[x1^2+y1^2]
/Sqrt[({x1,y1}-{x2,y2}).({x1,y1}-{x2,y2})],{f,0,2 Pi}]
{x2,-10,10},{y2,-10,10},PlotPoints->50]
```



```
Out[2] = - ContourGraphics -
```

### 25.3. Основное уравнение электронном оптики

Траектория электрона  $r = r(z)$  в аксиально-симметричном поле с потенциалом  $\Phi = \Phi(r, z)$  при движении на малом расстоянии от оси  $z$  и под небольшими углами к ней описывается уравнением:

$$\frac{d^2 r}{dz^2} + \frac{1}{2\Phi} \frac{d\Phi}{dz} \frac{dr}{dz} + \frac{1}{4\Phi} \frac{d^2 \Phi}{dz^2} r = 0, \quad (9)$$

где можно считать, что потенциал зависит только от  $z$ . Показать, что для любого  $\Phi(z)$ , для которого  $\Phi''(z) > 0$ , поле будет фокусировать электронный пучок, и траектории с разными начальными условиями для  $r(0)$  пересекают ось  $z$  в одной точке.

В качестве примера рассмотрим потенциал  $\Phi(z) = 1 + 2e^z$ .

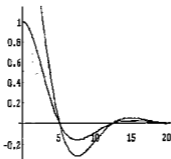
```
In[1] := f=1+2 Exp[z];
```

```
In[2] := eq=r''[z]+D[f,z]/2/f*r'[z]+D[f,z,z]/4/f*r[z]==0;
```

```
In[3] := w1=NDSolve[{eq,r[0]==1,r'[0]==0},r[z],{z,0,20}];
```

```
In[4] := w2=NDSolve[{eq,r[0]==2,r'[0]==0},r[z],{z,0,20}];
```

```
In[5] := Plot[{r[z]/.w1,r[z]/.w2},{z,0,20}]
```



```
Out[5] = - Graphics -
```

## 26. Решение уравнения Шредингера

Уравнение Шредингера — одно из фундаментальных уравнений квантовой физики. Описание двухчастичных нерелятивистских связанных состояний представляет значительный интерес в физике молекул, атомов, ядер и элементарных частиц. К сожалению, точные аналитические решения этого уравнения могут быть найдены только в ограниченном числе случаев, которые традиционно рассматриваются в учебниках по квантовой механике. Известны различные численные методы решения уравнения Шредингера для связанных состояний. Ниже будет рассмотрена программа “schroe”, разработанная Францем Шоберлем (Franz.Schoeberl@Univie.ac.at) для пакета Mathematica, которая позволяет найти собственные значения энергии и собственные функции численными методами.

В рамках нерелятивистской потенциальной модели взаимодействия бесспиновых частиц описывается сферически-симметричным потенциалом  $V(r)$ , где  $r$  — расстояние между частицами. В системе единиц  $\hbar = c = 1$  уравнение Шредингера для двух бесспиновых частиц с массами  $m_1$  и  $m_2$  записывается в виде:

$$\left(-\frac{\nabla^2}{2\mu} + V(r) + \frac{l(l+1)}{2\mu r^2}\right) \Psi_{n,l,m}(\vec{r}) = E_{n,l} \Psi_{n,l,m}(\vec{r}), \quad (10)$$

где  $\mu = m_1 m_2 / (m_1 + m_2)$  — приведенная масса системы,  $E_{n,l}$  — энергия связи, отвечающая главному квантовому числу  $n = 1, 2, \dots$ , орбитальному квантовому числу  $l = 0, 1, \dots, n-1$  и не зависящая от магнитного квантового числа  $m = -l, \dots, -1, 0, 1, \dots, l$ . В случае сферической симметрии потенциала взаимодействия  $V(r)$  решение уравнения (1) в сферических координатах представляется в виде

$$\Psi_{n,l,m}(\vec{r}) = \frac{U_{n,l}(r)}{r} Y_{l,m}(\theta, \phi), \quad (11)$$

где  $Y_{l,m}(\theta, \phi)$  — хорошо известные сферические функции. После подстановки (2) в уравнение (1) получаем уравнение для  $U_{n,l}$ :

$$\left[ \frac{1}{2\mu} \left( -\frac{d^2}{dr^2} + \frac{l(l+1)}{r^2} \right) + V(r) \right] U_{n,l}(r) = E_{n,l} U_{n,l}(r). \quad (12)$$

Уравнение (12) решается программой “schroe”. Перед запуском основного модуля необходимо ввести потенциал взаимодействия  $V(r)$ :

In[1] := vl[x\_] := ... (явный вид потенциала).

При этом предполагается, что все величины имеют одну размерность. Например, 1 эВ (электрон-вольт) или кратную ему 1 кэВ, 1 МэВ и т.д.

Затем ищется значение координаты, в которой эффективная потенциальная энергия ( $V_{eff}(r) = V(r) + l(l+1)/2\mu r^2$ ) принимает минимальное значение:

```
In[2] := xwmin2[m1_, m2_, l_] := FindMinimum[2*m1*m2/(m1+m2)
      *vl[xri]+l*(l+1)/xri^2, {xri, 0.5}]
In[3] := N[xwmin2[m1, m2, l]]
```

Здесь  $m_1, m_2$  и  $l$  — конкретные численные значения.

Программа решения приведенного уравнения (12) запускается командой:

```
In[4] := schroe[xri, e1, eu, n0, l, h, m1, m2],
```

где  $x_i$  — значение, найденное в предыдущем операторе,  $e_1$  и  $e_u$  — нижняя и верхняя границы отрезка, на котором ищется собственное значение энергии  $E_{n,l}$ ,  $n_0 = n_r = n - l - 1$  — радиальное квантовое число ( $n_r = 0, 1, \dots$ ),  $h$  — шаг интегрирования, который задает точность определения значения  $E_{n,l}$ ,  $m_1$  и  $m_2$  — массы частиц в системе. Описанный выше синтаксис команды "schroe" используется в системе Mathematica 2.2, в случае Mathematica 3.0 не нужно определять точку минимума эффективного потенциала, и команда "schroe" не содержит первого аргумента "x\_i".

В качестве примера рассмотрим классическую задачу о спектре энергий атома позитрония — системы, состоящей из электрона ( $e^-$ ) и позитрона ( $e^+$ ). В нерелятивистском приближении потенциал взаимодействия электрона и позитрона чисто кулоновский

$$V(r) = -\frac{\alpha}{r}, \quad (13)$$

где  $\alpha = e^2/(4\pi\hbar c) = 1/137.035989$  — постоянная тонкой структуры. Известно, что энергия основного состояния позитрония составляет несколько эВ, поэтому массу электрона удобно выразить также в эВ:

$m = m_1 = m_2 = 510999.07$  эВ. Выбор численного значения параметра  $h$  связан с точностью определения энергии  $\delta E$ .

Итак, определим потенциал (4) и оператор `xwmin2`:

```
In[1] := vl[x_] := -1./137.035989/x
In[2] := xwmin2[m1_, m2_, l_] := FindMinimum[2*m1*m2/(m1+m2)*
    vl[xri]+1*(1+1)/xri^2, {xri, 0.5}]
```

Следующая команда находит собственные значения и собственные функции, отвечающие  $n_r = 0$  и  $l = 0$  при значении  $h = 10^{-6}$  эВ<sup>-1</sup>, а также выводит на печать  $h$  и соответствующее собственное значение энергии:

```
In[3] := (h=0.000001;m1=510999.07;m2=510999.07;l=0;n0=0;
    N[xwmin2[m1,m2,l]];
    schroe[xri,-10,0,n0,l,h,m1,m2])
```

Результаты расчета выводятся в следующем виде:

```
E = -6.80284977, L = 0, N = 0, Integrationsteps =
5923, h = 1. 10-6, del = 1. 10-7, el = -10, eu = 0,
Largest x, upper integration limit, XMAX =
0.0059221, Smallest x, lower integration limit, XMIN\
= del = 1. 10-7. The reduced not normalized wave
function is yschr[x]. The normalization factor is given
by: 1/N Integrate[yschr[x]^2, {x, del, xmax}]
```

Затем программа рисует график ненормированной приведенной волновой функции  $U_{n,l}(\tau)$  (рис. 2).

Результаты расчетов при различных значениях  $h$  представлены в табл. 9:

Таблица 9

$h, \text{эВ}^{-1}$	$E_{1,0}, \text{эВ}$	N
$10^{-2}$	-6.35272502	1
$10^{-3}$	-3.72222423	9
$10^{-4}$	-6.78926944	59
$10^{-5}$	-6.80275440	633
$10^{-6}$	-6.80284977	5923

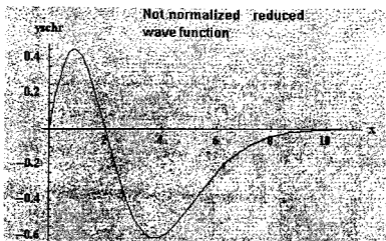


Рис. 2

В табл. 9  $N$  — число шагов интегрирования. Точность  $10^{-6}$  эВ собственной энергии основного состояния позитрония ( $E_{1,0} = -m\alpha^2/4 = -6.802849$  эВ) достигается при  $h = 10^{-6}$  эВ $^{-1}$  и числе шагов интегрирования  $N=5923$ .

Нормировочный коэффициент вычисляется следующей командой:

```
In[4] := cnorm:=1./NIntegrate[yschr[x]^2,{x,del,xmax}],
```

где  $del$  и  $xmax$  — определенные в программе минимальное и максимальное значения координаты, при которых задаются граничные условия для приведенной волновой функции  $yschr[x]$ , т.е.  $yschr[del] = y_{schr}[xmax]=0$ . После определения нормировочной константы можно рассчитывать средние значения функций  $\langle r^k \rangle$ , через которые выражаются основные характеристики позитрония. Например, среднеквадратичный радиус позитрония  $\langle r^2 \rangle$  определяется командой:

```
In[4] := Sqrt[cnorm*NIntegrate[x^2 yschr[x]^2,{x,del,xmax}]]
```



Расчет, выполненный при  $n_r = 0$ ,  $l = 0$  и  $h = 10^{-6}$  эВ<sup>-1</sup>, дает:

$$\begin{aligned} \langle r^2 \rangle &= 8.63031 \cdot 10^{-7} \text{ эВ}^{-2}, \\ \langle r \rangle &= 8.04525 \cdot 10^{-4} \text{ эВ}^{-1}, \\ \langle 1/r \rangle &= 1.86447 \cdot 10^3 \text{ эВ}, \\ \langle r^3 \rangle &= 6.94988 \cdot 10^6 \text{ эВ}^2. \end{aligned}$$

Используя связь  $1 \text{ эВ}^{-1} = 1.97327053 \cdot 10^{-5}$  см, можно выполнить пересчет от эВ к см.

Другим характерным примером использования программы "schgoe" является расчет спектра масс мезонов, состоящих из тяжелого кварка тяжелого антикварка:  $c\bar{c}$ ,  $b\bar{b}$ . Потенциал взаимодействия кварка и антикварка без учета спин-спинового и спин-орбитального взаимодействия представляется в виде суммы пертурбативной части, отвечающей одноглюонному обмену на малых расстояниях между кварком и антикварком, и непертурбативной линейно растущей с расстоянием части, обеспечивающей конфайнмент:

$$V(r) = -\frac{a}{r} + br - c, \quad (14)$$

где  $a = 0.52$ ,  $b = 0.182 \text{ ГэВ}^2$  и  $c = 0.87 \text{ ГэВ}$  при  $m_c = 1.84 \text{ ГэВ}$  и  $m_b = 5.18 \text{ ГэВ}$ . Добавки к потенциалу, учитывающие спин-орбитальное и спин-спиновое взаимодействие, имеют следующий вид:

$$V_{LS}(\vec{r}) = \frac{(\vec{L}\vec{S}_1)}{2m_1^2} \left( -\frac{dV(r)}{rdr} + \frac{8\alpha_s}{3r^3} \right) + \frac{4\alpha_s(\vec{L}\vec{S})}{3m_1m_2r^3}, \quad (15)$$

$$V_{SS}(\vec{r}) = \frac{8\alpha_s}{9m_1m_2} (\vec{S}_1\vec{S}_2) 4\pi\delta(\vec{r}) + \frac{4\alpha_s}{3m_1m_2r^3} [3(\vec{S}_1\vec{n})(\vec{S}_2\vec{n}) - (\vec{S}_1\vec{S}_2)], \quad (16)$$

где  $\vec{S} = \vec{S}_1 + \vec{S}_2$  — оператор полного спина системы,  $\vec{n} = \vec{r}/r$ ,  $m_1$  и  $m_2$  — массы кварков,  $\alpha_s = 0.31$  — константа кварк-глюонного взаимодействия.

Масса связанного состояния (мезона) определяется соотношением:

$$M = m_1 + m_2 + E_{n,l} + E_{SS} + E_{LS}, \quad (17)$$

где  $E_{n,l}$  — собственное значение энергии связи уравнения (12) с потенциалом  $V(r)$  (14). Значения энергий связи  $E_{SS}$  и  $E_{LS}$ , связанных со спин-спиновым и спин-орбитальным взаимодействием, много меньше  $E$  и могут быть рассчитаны по теории возмущений, т.е.

$$E_{SS} = \int \Psi_{n,l}^*(\vec{r}) V_{SS}(\vec{r}) \Psi_{n,l}(\vec{r}) d^3r, \quad (18)$$

$$E_{LS} = \int \Psi_{n,l}^*(\vec{r}) V_{LS}(\vec{r}) \Psi_{n,l}(\vec{r}) d^3r, \quad (19)$$

где  $\Psi_{n,l}(\vec{r})$  — собственные функции уравнения (1) с потенциалом (5). В самом простом случае состояний с  $l = 0$  ненулевое значение имеет только поправка к энергии, связанная со спин-спиновым взаимодействием:

$$E_{SS} = \frac{32}{9} \pi \alpha_s \frac{|\Psi_{n,l}(0)|^2}{m_1 m_2} \langle \vec{S}_1 \vec{S}_2 \rangle, \quad (20)$$

где

$$|\Psi_{n,l}(0)|^2 = \int \Psi_{n,l}^*(\vec{r}) \delta(\vec{r}) \Psi_{n,l}(\vec{r}) d^3r,$$

$$\langle \vec{S}_1 \vec{S}_2 \rangle = \begin{cases} -3/4 & \text{для } S = 0, \\ +1/4 & \text{для } S = 1. \end{cases}$$

Значение волновой функции в нуле  $\Psi_{n,l}(0)$  при  $l = 0$  может быть выражено через среднее значение производной потенциала  $V(r)$ :

$$|\Psi_{n,l}(0)|^2 = \frac{m_1 m_2}{2\pi(m_1 + m_2)} \left\langle \frac{dV(r)}{dr} \right\rangle, \quad (21)$$

$$\left\langle \frac{dV(r)}{dr} \right\rangle = \int \Psi_{n,l}^*(\vec{r}) \frac{dV(r)}{dr} \Psi_{n,l}(\vec{r}) d^3r = \int_0^\infty U_{n,l}^2(r) \frac{dV(r)}{dr} dr.$$

В качестве примера выполним расчет масс S-волновых ( $l = 0$ ) состояний системы, состоящей из  $c$ - и  $\bar{c}$ -кварков.

```
In[1] := vl[x_] := -0.52/x + 0.182*x - 0.87
In[2] := xwmin2[m1_, m2_, l_] := FindMinimum[2*m1*m2/(m1+m2)
      *vl[xri] + l*(1+1)/xri^2, {xri, 0.5}]
In[3] := {h=0.01; m1=1.84; m2=1.84; l=0; n0=0; als=0.31;
      N[xwmin2[m1, m2, l]];
      schroe[xri, -1, 0, n0, l, h, m1, m2]}
```

```

In[4] := cnorm:=1./NIntegrate[yschr[x]^2,{x,del,xmax}]
In[5] :=(pot=vl[q];dpot=D[pot,q]);
In[6] := psi = cnorm*m1*m2/(2*Pi*(m1+m2))*
        NIntegrate[dpot*yschr[q]^2,{q,del,xmax}]
In[7] := ess=N[32/9*als*Pi/(m1*m2)*psi]
In[8] := mv=m1+m2+N[ep/ww]+1/4*ess
In[9] :=ms=m1+m2+N[ep/ww]-3/4*ess

```

Здесь:  $\psi$  - квадрат модуля волновой функции в нуле,  $m_v$  - масса векторного состояния ( $S=1$ ),  $m_s$  - масса псевдоскалярного состояния ( $S=0$ ),  $N[\text{ep}/\text{ww}]$  значение энергии связи  $E_{n,l}$ . Для двух нижних  $S$ -волновых состояний при  $\hbar=10^{-2}$  ГэВ<sup>-1</sup> получаются следующие значения (табл. 10).

Таблица 10

Состояние	$ \Psi(0) ^2$ , ГэВ <sup>3</sup>	Масса, ГэВ, расчет	Масса, ГэВ, эксперимент
$1^3S_1$	0.1157	3.096	$3.09688 \pm 0.00004$
$1^1S_0$	0.1157	2.977	$2.9798 \pm 0.0021$
$2^3S_1$	0.0737	3.675	$3.6860 \pm 0.0001$
$2^1S_0$	0.0737	3.599	нет

## 27. Вычисления в квантовой теории поля

### 27.1. Поляризационные эффекты в комптоновском рассеянии

Рассмотрим процесс рассеяния линейно поляризованных фотонов на неполяризованных электронах. Поляризации конечных частиц не измеряются. Процесс  $\gamma e \rightarrow \gamma e$  описывается двумя диаграммами Фейнмана (рис. 3), которым отвечают амплитуды:

$$\mathcal{M}_1 = \frac{e^2 \varepsilon_\mu^*(k_2) \varepsilon_\nu(k_1)}{(k_1 + p_1)^2 - m^2} \bar{U}(p_2, s_2) \gamma_\mu (\hat{k}_1 + \hat{p}_1 + m) \gamma_\nu U(p_1, s_1), \quad (22)$$

$$\mathcal{M}_2 = \frac{e^2 \varepsilon_\mu^*(k_2) \varepsilon_\nu(k_1)}{(p_1 - k_2)^2 - m^2} \bar{U}(p_2, s_2) \gamma_\nu (\hat{p}_1 - \hat{k}_2 + m) \gamma_\mu U(p_1, s_1). \quad (23)$$



Рис. 3

Находим  $\mathcal{M}_1^*$  и  $\mathcal{M}_2^*$ , затем  $|\mathcal{M}_1|^2$ ,  $|\mathcal{M}_2|^2$  и  $\mathcal{M}_1 \mathcal{M}_2^*$ . Суммируем по поляризациям конечных частиц и усредняем по поляризациям начального электрона, используя следующие соотношения:

$$\begin{aligned} \sum_s \varepsilon_\mu^*(k_2) \varepsilon_\nu(k_2) &= -g_{\mu\nu}, \\ \sum_s U(p_2, s_2) \bar{U}(p_2, s_2) &= \hat{p}_2 + m, \\ \sum_s U(p_1, s_1) \bar{U}(p_1, s_1) &= \hat{p}_1 + m. \end{aligned} \quad (24)$$

Поляризационное состояние начальных фотонов описывается матрицей плотности, которая в случае линейно поляризованных фотонов имеет вид:

$$\rho_{\mu\nu} = \varepsilon_{\mu}^*(k_1)\varepsilon_{\nu}(k_1) = \frac{1}{2} \left( (1 + \xi_3)\varepsilon_{\mu}^{(1)}\varepsilon_{\nu}^{(1)} + (1 - \xi_3)\varepsilon_{\mu}^{(2)}\varepsilon_{\nu}^{(2)} \right), \quad (25)$$

где  $\varepsilon_{\mu}^{(1)}$  и  $\varepsilon_{\mu}^{(2)}$  – базисные 4-векторы, которые ортогональны друг к другу и 4-импульсу фотона. В кулоновской калибровке можно положить  $\varepsilon^{(i)} = (0, \vec{\varepsilon}^{(i)})$ . Выберем в качестве базисных единичные векторы  $\vec{\varepsilon}^{(1)} = (1, 0, 0)$  и  $\vec{\varepsilon}^{(2)} = (0, 1, 0)$ , направленные вдоль осей X и Y (ось Z направлена по импульсу фотона  $\vec{k}_1$ ).

$$|\mathcal{M}|^2 = |\mathcal{M}_1|^2 + |\mathcal{M}_2|^2 + 2\mathcal{M}_1\mathcal{M}_2^*, \quad (26)$$

где

$$\begin{aligned} |\overline{\mathcal{M}_1}|^2 &= \frac{-e^4 \rho_{\mu\nu}}{8(p_1 k_1)^2} \text{Tr}((\hat{p}_2 + m)\gamma_{\sigma}(\hat{p}_1 + \hat{k}_1 + m)\gamma_{\mu}(\hat{p}_1 + m) \\ &\quad \gamma_{\nu}(\hat{p}_1 + \hat{k}_1 + m)\gamma_{\sigma}), \end{aligned} \quad (27)$$

$$\begin{aligned} |\overline{\mathcal{M}_2}|^2 &= \frac{-e^4 \rho_{\mu\nu}}{8(p_1 k_2)^2} \text{Tr}((\hat{p}_2 + m)\gamma_{\mu}(\hat{p}_1 - \hat{k}_2 + m)\gamma_{\sigma}(\hat{p}_1 + m) \\ &\quad \gamma_{\sigma}(\hat{p}_1 - \hat{k}_2 + m)\gamma_{\nu}), \end{aligned} \quad (28)$$

$$\begin{aligned} \overline{\mathcal{M}_1\mathcal{M}_2^*} &= \frac{e^4 \rho_{\mu\nu}}{8(p_1 k_1)(p_1 k_2)} \text{Tr}((\hat{p}_2 + m)\gamma_{\sigma}(\hat{k}_1 + \hat{p}_1 + m) \\ &\quad \gamma_{\sigma}(\hat{p}_1 - \hat{k}_2 + m)\gamma_{\nu}). \end{aligned} \quad (29)$$

Дифференциальное сечение процесса в системе координат, где начальный электрон покоился и вектор рассеянного фотона  $\vec{k}_2$  лежит в плоскости XZ (это фиксирует азимутальный угол рассеяния  $\phi = 0$ ), имеет вид:

$$\frac{d\sigma}{d \cos \vartheta d\phi} = \frac{w_2^2}{64\pi^2 m^2 w_1^2} |\overline{\mathcal{M}}|^2. \quad (30)$$

Выразим скалярные произведения 4-векторов, входящих в задачу, через энергию начального ( $w_1$ ) и конечного ( $w_2$ ) фотонов, угол рассеяния фотона ( $\vartheta$ ):

$$(\varepsilon^{(1)} k_1) = (\varepsilon^{(2)} k_1) = 0,$$

$$\begin{aligned}
(\varepsilon^{(1)} p_1) &= (\varepsilon^{(2)} p_1) = 0, \\
(\varepsilon^{(1)} p_2) &= (\varepsilon^{(2)} p_2) = 0, \\
(\varepsilon^{(1)} k_2) &= -w_2 \sin \vartheta, \\
(\varepsilon^{(2)} k_2) &= -w_2 \sin \vartheta, \\
(k_1 p_1) &= m w_1, \\
(k_2 p_1) &= m w_2, \\
(k_1 k_2) &= w_1 w_2 (1 - \cos \vartheta).
\end{aligned}
\tag{31}$$

Ниже приведена программа для расчета дифференциального сечения рассеяния линейно поляризованных фотонов ( $\xi_3 \neq 0$ ) на электронах.

```

In[1] := << d:\wnmath22\packages\feynccalc\feynccalc.m

In[2] := {ScalarProduct[e1,k1]=0; ScalarProduct[e2,k1]=0; \
  ScalarProduct[e1,e1]=0; ScalarProduct[e1,e1]=-1; \
  \ ScalarProduct[e2,e2]=-1; ScalarProduct[k1,k1]=0; \
  \ ScalarProduct[k2,k2]=0; \
  \ ScalarProduct[k2,k1]=w1*w2*(1-Cos[theta]); \
  \ ScalarProduct[k1,p1]=m*w1; \ ScalarProduct[k2,p1]=m*w2; \
  \ ScalarProduct[e1,k2]=-w2*Sin[theta]; \
  \ ScalarProduct[e2,k2]=0; \
  \ ScalarProduct[e1,p1]=0; \ ScalarProduct[e2,p1]=0;}

In[3] := p2=p1+k1-k2;

In[4] := (A1=DiracSlash[p1]+m; A2=DiracSlash[p2]+m; \
  A11=DiracSlash[p1+k1]+m; A12=DiracSlash[p1-k2]+m; \
  Si=DiracMatrix[si]; Mu=DiracMatrix[mu]; \
  Nu=DiracMatrix[nu];)

In[5] := Rmm=1/2*((1+x3)*FourVector[e1,mu]*FourVector[e1,nu]+ \
  (1-x3)*FourVector[e2,mu]*FourVector[e2,nu]);

```

```

In[6] := M11=Contract[-Rmn/(8*ScalarProduct[p1,k1]^2)*
      Tr[A2.Si.A11.Mu.A1.Nu.A11.Si]];

In[7] := M22=Contract[-Rmn/(8*ScalarProduct[p1,k2]^2)*
      Tr[A2.Mu.A12.Si.A1.Si.A12.Nu]];

In[8] := M12=Contract[Rmn/(8*ScalarProduct[p1,k1]*
      ScalarProduct[p1,k2])*Tr[A2.Si.A11.Mu.A1.Si.A12.Nu]];

In[9] := MM=Together[M11+M22+2*M12];

In[10] := DS=Simplify[w2^2*MM/(64*Pi^2*m^2*w1^2)];

In[11] := (DS=DS*m^2; DS=DS/.m->w1*w2*(1-Cos[theta])/(w1-w2);
      DS=Simplify[DS/m^2]; DS=Factor[DS];)

In[12] := DS=DS*(4*Pi*alf)^2

Out[12] = (alf m w1 w2 (w1^2 + w2^2 - w1 w2 Sin[theta] -
      w1 w2 x3 Sin[theta] )) / 2

```

Таким образом,

$$\frac{d\sigma}{d\cos\vartheta d\phi} = \frac{\alpha^2 w_2}{2m^2 w_1^3} (w_1^2 + w_2^2 - w_1 w_2 \sin^2\vartheta (1 + \xi_3)). \quad (32)$$

Видно, что сечение рассеяния фотонов, поляризованных перпендикулярно плоскости рассеяния ( $\xi_3 = -1$ ), больше, чем для фотонов, поляризованных в плоскости рассеяния XZ ( $\xi_3 = 1$ ). При  $\xi_3 = 0$  получаем известный ответ для сечения рассеяния неполяризованных фотонов.

## 27.2. Распад $\mu \rightarrow e\bar{\nu}_e\nu_\mu$

Распад мюона  $\mu \rightarrow e\bar{\nu}_e\nu_\mu$  является чисто лептонным процессом, который может быть рассчитан до конца в рамках теории возмущений. С другой стороны, этот процесс наиболее тщательно изучен на опыте и это позволяет провести проверку полученных теоретических результатов с большой точностью.

Слабые процессы при низких энергиях, когда массы промежуточных векторных бозонов  $M_W, M_Z$  много больше энергий и масс частиц, участвующих в реакции, хорошо описываются в рамках эффективной модели четырехфермионного взаимодействия. Амплитуда распада мюона представляется в виде произведения двух левых токов:

$$\mathcal{M} = \frac{G_F}{\sqrt{2}} \bar{U}(q_2) \gamma^\alpha (1 - \gamma_5) U(p) \bar{U}(k) \gamma_\alpha (1 - \gamma_5) V(q_1), \quad (33)$$

где  $p$  — четырехимпульс мюона,  $k$  — четырехимпульс электрона,  $q_1$  — четырехимпульс электронного антинейтрино,  $q_2$  — четырехимпульс мюонного нейтрино,  $G_F$  — эффективная константа четырехфермионного взаимодействия Ферми ( $G_F \cong 10^{-5}/m_p^2$ , где  $m_p$  — масса протона). Сопряженная амплитуда

$$\mathcal{M}^* = \frac{G_F}{\sqrt{2}} \bar{U}(p) \gamma^\beta (1 - \gamma_5) U(q_2) \bar{V}(q_1) \gamma_\beta (1 - \gamma_5) U(k). \quad (34)$$

Квадрат модуля амплитуды, просуммированной по поляризациям конечных частиц и усредненный по поляризациям начального мюона, приводится к виду:

$$\begin{aligned} |\overline{\mathcal{M}}|^2 = & \frac{G_F^2}{4} \text{Tr} [\hat{q}_2 \gamma^\alpha (1 - \gamma_5) (\hat{p} + M) (1 + \gamma_5) \gamma^\beta] \times \\ & \text{Tr} [(\hat{k} + m) \gamma_\alpha (1 - \gamma_5) \hat{q}_1 (1 + \gamma_5) \gamma_\beta], \end{aligned} \quad (35)$$

где  $m$  — масса электрона,  $M$  — масса мюона.

Дифференциальная ширина распада, в системе покоя мюона, связана с квадратом модуля амплитуды соотношением

$$d\Gamma = \frac{|\overline{\mathcal{M}}|^2}{2M} d\Phi, \quad (36)$$



где  $d\Phi$  инвариантный фазовый объем конечных частиц:

$$d\Phi = (2\pi)^4 \delta^{(4)}(p - k - q_1 - q_2) \frac{d^3k}{2E_k(2\pi)^3} \frac{d^3q_1}{2E_1(2\pi)^3} \frac{d^3q_2}{2E_2(2\pi)^3}.$$

В системе координат, в которой ось  $z$  направлена по импульсу электрона и вектор электронного антинейтрино лежит в плоскости  $\{xz\}$ , четыре импульсы частиц равны:

$$\begin{aligned} p &= (M, 0, 0, 0), & q_1 &= (E_1, E_1 \sin \theta_1, 0, E_1 \cos \theta_1), \\ k &= (E_k, 0, 0, k), & q_2 &= p - k - q_1. \end{aligned}$$

Здесь

$$\cos \theta_1 = \frac{M^2 + m^2 - 2ME_k - 2ME_1 + 2E_1E_k}{2kE_1}.$$

С учетом этого после стандартных преобразований фазовый объем приводится к виду

$$d\Phi = -\frac{dE_k dE_1}{32\pi^3}.$$

Чтобы получить полную ширину распада, необходимо проинтегрировать дифференциальную ширину распада по  $E_1$  и  $E_k$  в пределах:

$$\frac{M^2 - 2ME_k + m^2}{2(M - E_k + k)} \leq E_1 \leq \frac{M^2 - 2ME_k + m^2}{2(M - E_k + k)},$$

$$m \leq E_k \leq \frac{M^2 + m^2}{2M}.$$

Ниже приведен текст программы, рассчитывающей полную ширину распада мюона:

```
In[1] := <<d:\wnmath22\packages\feyncalc\feyncalc.m;
```

```
In[2] := (DS[p_,m_] := DiracSlash[p] + m; DM[a_] := DiracMatrix[a];
D5 := DiracMatrix[5]);
```

```
In[3] := (fe = Tr[DS[q2,0].DM[a].(1-D5).DS[p,M].(1+D5).DM[be]]);
```

fm=Tr[DS[k,m].DM[a1].(1-D5).DS[q1,0].(1+D5).DM[be]]];

In[4] := F=GF^2/4\*Simplify2[Contract[fe\*fm,EpsContract->True]]

Out[4] = 64 GF<sup>2</sup> k.q2 p.q1

In[5] := (ScalarProduct[p,q1]=M\*E1;ScalarProduct[k,q2]=  
(M^2-m^2-2\*M\*E1\*M)/2);

In[6] := E1max=(M^2+m^2-2\*M\*EK)/2/(M-K-EK);

In[7] := E1min=(M^2+m^2-2\*M\*EK)/2/(M+K-EK);

In[8] := DF=-F/(2\*M\*32\*Pi^3)

Out[8] = 
$$\frac{-(E1 GF^2 (-m^2 - 2 E1 M + M^2))}{3 \cdot 2 \text{ Pi}}$$

In[9] := DF1=Simplify2[Integrate[DF,{E1,E1min,E1max}]]];

In[10] := DF1=Simplify2[DF1/.{K^2->EK^2-m^2,K->Sqrt[EK^2-m^2]}]

Out[10] = 
$$\frac{-(GF^2 \text{ Sqrt}[EK - m] \text{ Sqrt}[EK + m] (m^2 - 2 EK M + M^2) (3 EK m^2 - 4 EK M^2 - 2 m^2 M + 3 EK M))}{3 (12 (EK - \text{Sqrt}[EK - m] \text{ Sqrt}[EK + m] - M))}$$

$$(EK + \text{Sqrt}[EK - m] \text{Sqrt}[EK + m] - M) \text{Pi}$$

In[11] := DF2=Simplify2[Integrate[DF1,{EK,m,(M^2+m^2)/(2\*M)}]]];

In[12] := DF2=Simplify2[DF2/.m->x\*M]

$$\text{Out}[12] = \frac{\text{GF} M^5 (1 - 8x^2 + 8x^6 - x^8 - 24x \text{Log}[x])}{192 \text{Pi}}$$

Таким образом,

$$\Gamma(\mu \rightarrow e\hat{\nu}_e\nu_\mu) = \frac{G_F^2 M^5}{192\pi^3} (1 - 8x^2 + 8x^6 - x^8 - 24x^4 \ln x),$$

где  $x = m/M$ .

### 27.3. Процесс $gg \rightarrow Q\bar{Q}$

В данном разделе вычисляется полное сечение рождения кварка и антикварка двумя глюонами, которое в лидирующем порядке по  $\alpha_s$  описывается тремя диаграммами (рис. 4).

$$\mathcal{M}_1 = g_s^2 \varepsilon_\mu^a(q_1) \varepsilon_\mu^b(q_2) T^a T^b \bar{U}(p_1) \gamma^\mu \frac{\hat{p}_1 - \hat{q}_1 + M}{(p_1 - q_1)^2 - M^2} \gamma^\nu V(p_2), \quad (37)$$

$$\mathcal{M}_2 = g_s^2 \varepsilon_\mu^a(q_1) \varepsilon_\mu^b(q_2) T^b T^a \bar{U}(p_1) \gamma^\nu \frac{\hat{p}_1 - \hat{q}_2 + M}{(p_1 - q_2)^2 - M^2} \gamma^\mu V(p_2), \quad (38)$$

$$\mathcal{M}_3 = g_s^2 \varepsilon_\mu^a(q_1) \varepsilon_\mu^b(q_2) T^c i f^{abc} \bar{U}(p_1) \frac{\gamma_\lambda}{(q_1 + q_2)^2} V(p_2) \times \\ \times V^{\mu\nu\lambda}(q_1 + q_2, -q_1, -q_2), \quad (39)$$

где

$$V^{\mu\nu\lambda}(p, q, k) = [(p - q)_\nu g_{\mu\lambda} + (q - k)_\lambda g_{\mu\nu} + (k - p)_\mu g_{\nu\lambda}].$$

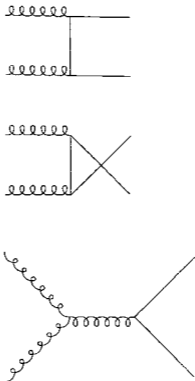


Рис. 4

Для суммирования по спиновым и цветовым степеням свободы глюонов воспользуемся соотношением полноты в фейнмановской калибровке:

$$\sum \epsilon_{\mu}^a(q) \epsilon_{\nu}^{*b}(q) = -\delta^{ab} g_{\mu\nu}. \quad (40)$$

Для исключения нефизических степеней свободы, дающих вклад в квадрат модуля амплитуды  $\mathcal{M}_3$  с трехглюонной вершиной, к квадрату модуля  $|\mathcal{M}_3|^2$  необходимо добавить квадрат модуля амплитуды  $|\mathcal{M}_4|^2$ , в которой глюонная петля заменена на петлю духов, обращающую в ноль вклад нефизических состояний.

Другой способ исключения вклада нефизических степеней свободы заключается в использовании “физической” калибровки, в которой

$$\sum \varepsilon_\mu^a(q_1) \varepsilon_\nu^{*b}(q_1) = -\delta^{ab} \left( g_{\mu\nu} - \frac{q_{1,\mu} q_{2,\nu} + q_{1,\nu} q_{2,\mu}}{q_1 \cdot q_2} \right), \quad (41)$$

$$\sum \varepsilon_\mu^a(q_2) \varepsilon_\nu^{*b}(q_2) = -\delta^{ab} g_{\mu\nu}. \quad (42)$$

Дальнейшие вычисления будут проводиться в фейнмановской калибровке. Выпишем выражения для квадратов модулей амплитуд, отвечающих отдельным диаграммам, интерференционные члены и соответствующие цветовые факторы  $C_{ij}$ :

$$|\overline{\mathcal{M}}_1|^2 = \frac{1}{4} g_s^2 C_{11} \text{Tr} [(\hat{p}_1 + M) \gamma^\mu (\hat{p}_1 - \hat{q}_1 + M) \gamma^\nu (\hat{p}_2 - M) \gamma_\nu (\hat{p}_1 - \hat{q}_1 + M) \gamma_\mu], \quad (43)$$

$$C_{11} = \frac{1}{64} \text{Tr} [T^a T^b T^b T^a].$$

$$|\overline{\mathcal{M}}_2|^2 = \frac{1}{4} g_s^2 C_{22} \text{Tr} [(\hat{p}_1 + M) \gamma^\nu (\hat{p}_1 - \hat{q}_2 + M) \gamma^\mu (\hat{p}_2 - M) \gamma_\mu (\hat{p}_1 - \hat{q}_2 + M) \gamma_\nu], \quad (44)$$

$$C_{22} = \frac{1}{64} \text{Tr} [T^b T^a T^a T^b].$$

$$2\overline{\mathcal{M}}_1 \overline{\mathcal{M}}_2^* = \frac{1}{2} g_s^2 C_{12} \text{Tr} [(\hat{p}_1 + M) \gamma^\mu (\hat{p}_1 - \hat{q}_1 + M) \gamma^\nu (\hat{p}_2 - M) \gamma_\mu (\hat{p}_1 - \hat{q}_2 + M) \gamma_\nu], \quad (45)$$

$$C_{12} = \frac{1}{64} \text{Tr} [T^a T^b T^a T^b].$$

$$|\overline{\mathcal{M}}_3|^2 = \frac{1}{4} g_s^2 C_{33} \text{Tr} [(\hat{p}_1 + M) \gamma_\lambda (\hat{p}_2 - M) \gamma^\rho] \times \\ \times V^{\mu\nu\lambda}(q_1 + q_2, -q_1, -q_2) V_{\mu\nu\rho}(-q_1 - q_2, q_1, q_2), \quad (46)$$

$$C_{33} = \frac{1}{64} \text{Tr} [T^c T^d] f^{abc} f^{abd}.$$

$$|\overline{\mathcal{M}}_4|^2 = 2 \frac{1}{4} g_s^2 C_{44} \text{Tr} [(\hat{p}_1 + M) \hat{q}_2 (\hat{p}_2 - M) \hat{q}_1], \quad (47)$$

$$C_{44} = \frac{1}{64} \text{Tr} [T^b T^d] f^{abc} f^{adc}.$$

$$2\overline{\mathcal{M}_3\mathcal{M}_1^*} = \frac{1}{2}g_s^2 C_{13} \text{Tr} [(\hat{p}_1 + M)\gamma^\lambda(\hat{p}_2 - M)\gamma_\nu(\hat{p}_1 - \hat{q}_1 + M)\gamma_\mu] \times \\ \times V^{\mu\nu\lambda}(q_1 + q_2, -q_1, -q_2), \quad (48)$$

$$C_{13} = \frac{1}{64} \text{Tr} [T^c T^b T^a] i f^{abc}.$$

$$2\overline{\mathcal{M}_3\mathcal{M}_2^*} = \frac{1}{2}g_s^2 C_{23} \text{Tr} [(\hat{p}_1 + M)\gamma^\lambda(\hat{p}_2 - M)\gamma_\mu(\hat{p}_1 - \hat{q}_2 + M)\gamma_\nu] \times \\ \times V^{\mu\nu\lambda}(q_1 + q_2, -q_1, -q_2), \quad (49)$$

$$C_{23} = \frac{1}{64} \text{Tr} [T^c T^a T^b] i f^{abc}.$$

Дифференциальное сечение рассеяния представлено в виде:

$$\frac{d\sigma}{dt} = \frac{1}{16\pi s^2} |\overline{\mathcal{M}}|^2, \quad (50)$$

где переменные Мандельштама определены следующим образом:  $s = (q_1 + q_2)^2$ ,  $t = (q_1 - p_1)^2$ ,  $u = (q_1 - p_2)^2$ . Полное сечение получается интегрированием по  $t$  в пределах

$$t_{max} = M^2 - s/2 + \sqrt{s(s - 4M^2)}/2,$$

$$t_{min} = M^2 - s/2 - \sqrt{s(s - 4M^2)}/2.$$

```
In[1] := <<d:\wnmath22\packages\feyncalc\feyncalc.m;
```

```
In[2] := SetMandelstam[s,t,u,q1,q2,-p1,-p2,0,0,M,M];
```

```
In[3] := (P1=DiracSlash[p1];Q2=DiracSlash[q2];
Q1=DiracSlash[q1];P2=DiracSlash[p2];
MU=DiracMatrix[mu];NU=DiracMatrix[nu]);
```

```

In[4] := A11=Simplify[Contract[Tr[(P1+M).MU.(P1-Q1+M).NU.
(P2-M).NU.(P1-Q1+M).MU]]]/(t-M^2)^2;

In[5] := A22=Simplify[Contract[Tr[(P1+M).NU.(P1-Q2+M).MU.
(P2-M).MU.(P1-Q2+M).NU]]]/(u-M^2)^2;

In[6] := A12=2*Simplify[Contract[Tr[(P1+M).MU.(P1-Q1+M).NU.
(P2-M).MU.(P1-Q2+M).NU]]]/((t-M^2)*(u-M^2));

In[7] := k=q1+q2;

In[8] := V=FourVector[-q1+q2,la]*MetricTensor[mu,nu]-
FourVector[k+q2,mu]*MetricTensor[la,nu]+
FourVector[k+q1,nu]*MetricTensor[mu,la];

In[9] := VC=FourVector[-q1+q2,la1]*MetricTensor[mu,nu]-
FourVector[k+q2,mu]*MetricTensor[la1,nu]+
FourVector[k+q1,nu]*MetricTensor[mu,la1];

In[10] := (LA=DiracMatrix[la];LAC=DiracMatrix[la1]);

In[11] := A33=Simplify[Contract[Tr[(P1+M).LA.(P2-M).LAC]*V
*VC]]/s^2;

In[12] := A44=Tr[(P1+M).Q2.(P2-M).Q1]/s^2;

In[13] := A13=2*Simplify[Contract[V*Tr[(P1+M).LA.(P2-M).NU.
(P1-Q1+M).MU]]]/s/(t-M^2);

In[14] := A23=2*Simplify[Contract[V*Tr[(P1+M).LA.(P2-M).MU.
(P1-Q2+M).NU]]]/s/(u-M^2);

```

In[15] := C11=GellMannTrace[GellMannMatrix[a,b,b,a]/2^4]/64  
Out[15] = 1/12

In[16] := C22=GellMannTrace[GellMannMatrix[b,a,a,b]/2^4]/64  
Out[16] = 1/12

In[17] := C12=GellMannTrace[GellMannMatrix[a,b,a,b]/2^4]/64  
Out[17] = -1/96

In[18] := C33=GellMannTrace[GellMannMatrix[c,c1]SU3F[a,b,c]  
SU3F[a,b,c1]]/2^2/64  
Out[18] = 3/16

In[19] := C44=GellMannTrace[GellMannMatrix[b,b1]/2^2  
SU3F[a,b,c]SU3F[a,b1,c]]/64  
Out[19] = 3/16

In[20] := C13=-I\*GellMannTrace[GellMannMatrix[c,b,a]  
SU3F[a,b,c]]/2^3/64  
Out[20] = 3/32

In[21] := C23=-I\*GellMannTrace[GellMannMatrix[c,a,b]/2^3  
SU3F[a,b,c]]/64  
Out[21] = -3/32

In[22] := FF=C11\*A11+C22\*A22+C33\*A33+C44\*A44+C12\*A12+  
C13\*A13+C23\*A23;

In[23] := FF=Simplify[FF/.u->-t-s+2\*M^2];

In[24] := tmax=M^2-s/2+L/2;



In[25] := tmin=M^2-s/2-L/2;

In[26] := TT=Together[Integrate[FF,{t,tmin,tmax}]];

In[27] := DEN=Expand[Denominator[TT]/.L->Sqrt[s\*(s-4\*M^2)]]

Out[27] = 
$$\frac{2}{-192 M^3 s}$$

In[28] := NUM=TT\*Denominator[TT];

In[29] := NUM=NUM/.{L^5->L\*s^2\*(s-4\*M^2)^2,L^3->L\*s\*(s-4\*M^2),  
L^2->s\*(s-4\*M^2)};

In[30] := NUM=NUM/.{(-L/2-s/2)->(L+s)/2,(L-s)/2->(s-L)/2};

In[31] := NUM=Collect2[NUM,{L,Log}]

Out[31]=

$$16 L^2 M^2 s^2 (31 M^2 + 7 s) + 64 M^2 s^4 (M^2 + 4 M^2 s + s^2) \log\left[\frac{-L + s}{2}\right] - 64 M^2 s^4 (M^2 + 4 M^2 s + s^2) \log\left[\frac{L + s}{2}\right]$$

In[32] := SECH=NUM/(16\*Pi\*s^2)\*(4\*Pi\*Als)^2/DEN;

In[33] := SECH2=Collect2[SECH/(Pi\*Als^2)\*3\*s^3,{L,Log}]

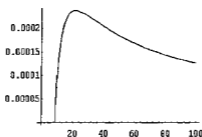
Out[33] =

$$\frac{L^4 (-31 M^2 - 7 s)}{4} + (-M^4 - 4 M^2 s - s^4) \log\left[\frac{-L + s}{2}\right] +$$

$$(M^4 + 4M^2s + s^2) \text{Log}\left[\frac{L+s}{s-L}\right]$$

```
In[34] := SECH3=0.4*SECH/.{L->Sqrt[s*(s-4*1.5^2)],M->1.5,
  Als->0.3};
```

```
In[35] := Plot[SECH3,{s,9,20},AxesLabel->{CrossSection [mbn],
  s[Gev**2]}]
```



```
Out[35] = - Graphics -
```

В операторе In[33] проводится деление на общий множитель  $\pi\alpha_s^2/3s^3$ , после чего сечение (SECH) элементарно приводится к виду:

$$\sigma(gg \rightarrow Q\bar{Q}) = \frac{\pi\alpha_s^2}{3s^3} \left[ (s^2 + 4sM^2 + M^4) \ln\left(\frac{s+L}{s-L}\right) - (7s + 31M^2)\frac{L}{4} \right], \quad (51)$$

где

$$L = \sqrt{s(s - 4M^2)}.$$

Затем строится кривая зависимости полного сечения  $\sigma(gg \rightarrow Q\bar{Q})$  от инварианта  $s$  при  $\alpha_s = 0.3$  и  $m_c = 1.5$  ГэВ.

# Предметный указатель

## Символы

- >, 47
- $\delta$ -функция Дирака, 62
  - многомерная, 62
- $\gamma$ , 12
- $\ln(x)$ , 16
- $\pi$ , 12
- $e^x$ , 16
- $\log_b(x)$ , 16
- Infinity, 11
- ., 81
- ?, 9, 10
- ?\*, 10
- ??, 9

## A

- Abs, 9, 14
- AbsolutePointSize, 32
- AbsoluteThickness, 32
- All, 30
- Apart, 44
- Append, 26
- Arg, 14
- Array, 27, 40
- AspectRatio, 30
- Automatic, 30
- Axes, 30
- AxesLabel, 30
- AxesOrigin, 30

## B

- Background, 30
- BarChart, 72
- BarChart3D, 75
- BernoulliDistribution, 56
- BesselJ, 16
- BetaDistribution, 56
- BinCounts, 53
- Binomial, 16
- BinomialDistribution, 56

## C

- CA, 90
- Cartesian, 68
- Catalan, 12
- CauchyDistribution, 56
- CDF, 57
- Ceiling, 14
- Cell, 8
- CentralMoment, 54
- CF, 90
- CForm, 49
- ChiDistribution, 56
- ChiSquareDistribution, 56
- Chop, 40
- Collect, 43
- ColorOutput, 30
- Column, 52
- ColumnDrop, 52

ColumnForm, 45  
 ColumnJoin, 52  
 ColumnTake, 52  
 CompoundExpression, 48  
 Complex, 11  
 ComplexInfinity, 11  
 ComplexToTrig, 60  
 Conjugate, 14, 80  
 ContourPlot, 35  
 Contract, 84  
 Coordinates, 69  
 CoordinateSystem, 69  
 Coordsys, 68  
 Cos, 9  
 CrossProduct, 67  
 CumulativeSums, 53  
 Curl, 70  
 Cylindrical, 68  
 CylindricalPlot3D, 72

**D**

D, 9, 41  
 Dashing, 32  
 DefaultFont, 30  
 Degree, 12  
 Det, 9, 40  
 Dimension, 79  
 DiracDelta, 62  
 DiracMatrix, 81, 89  
 DiracMatrix[5], 81, 89  
 DiracMatrix[6], 81, 89  
 DiracMatrix[7], 81, 89  
 DiracOrder, 85  
 DiracSimplify, 85  
 DiracSlash, 81, 89  
 DiracTrace, 86  
 Direction, 43  
 DiscreteUniformDistribution,  
     57  
 Div, 69  
 Do, 47  
 Domain, 57  
 DotProduct, 67  
 Drop, 23  
 DropNonNumeric, 53  
 DropNonNumericColumn, 53  
 DSolve, 47, 61

**E**

Eigensystem, 40  
 Eigenvalues, 9, 39  
 Eigenvectors, 39  
 EpsContract, 84  
 ErrorListPlot, 78  
 EulerGamma, 12  
 EvaluateDiracTrace, 86  
 Exp, 16  
 Exp[Fit[Log[...]]], 28  
 Expand, 43  
 ExpandScalarProduct, 84  
 ExponentialDistribution, 56

**F**

Factor, 43

FactorTerms, 43  
FermionSpinSum, 90  
FeynAmpDenominator, 83  
FindMinimum, 20  
FindRoot, 9, 20  
Fit, 28  
Floor, 14  
For, 48  
FortranForm, 49  
FourierCosTransform, 63  
FourierExpSeries, 64  
FourierSinTransform, 64  
FourierTransform, 63  
FourierTrigSeries, 64  
FourVector, 79, 80, 89  
Frame, 30  
FrameLabel, 31  
FrameStyle, 31  
FrameTicks, 31  
Frequencies, 53  
FV, 89

## G

GA, 89  
GA[5], 89  
GA[6], 89  
GA[7], 89  
GammaDistribution, 56  
GCD, 9  
GellMannMatrix, 81, 82  
GellMannTrace, 88  
GeneralizedBarChart, 73

GeometricDistribution, 57  
GeometricMean, 54  
GGV, 89  
GluonGhostVertex, 89  
GluonVertex, 90  
GoldenRatio, 12  
Goto, 48  
Grad, 69  
Graphics[Text[...]], 34  
GraphicsArray, 37  
GridLines, 31  
GS, 89  
GV, 90

## H

Help, 10

## I

I, 12  
If, 48  
Im, 14  
Indeterminate, 11  
Infinity, 11  
InputForm, 49  
Insert, 26  
Integer, 11  
Integrate, 9, 41  
InterpolatingFunction, 21, 28  
InterpolatingPolynomial, 28  
Interpolation, 28  
InterpolationOrder, 28  
Inverse, 39

- InverseFourierCosTransform[expr, w], 63
  - InverseFourierSinTransform, 64
  - InverseFourierTransform, 63
  - InverseLaplaceTransform, 66
- J**
- Join, 26
- L**
- LabeledListPlot, 77
  - LaguerreL, 16
  - LaplaceDistribution, 56
  - LaplaceTransform, 66
  - Laplacian, 70
  - Last, 23
  - LC, 89
  - Length, 23
  - LeptonSpinor, 82, 83
  - LeviCivita, 79, 84, 89
  - Limit, 43
  - LinearSolve, 40
  - ListContourPlot, 35
  - ListPlot, 32
  - ListPlot3D, 36
  - Log, 16
  - LogSeriesDistribution, 57
- M**
- MatrixForm, 26
  - Max, 14, 24
  - Mean, 54, 57
  - MeanDeviation, 54
- M**
- Median, 55
  - Method, 20
  - MetricTensor, 79, 80, 89
  - Min, 14, 24
  - Mod, 16
  - Momentum, 79
  - MT, 89
- N**
- N, 12, 13, 16
  - NDSolve, 21
  - NegativeBinomialDistribution, 57
  - Nest, 48
  - NFourierExpSeries, 65
  - NFourierTransform, 63
  - NFourierTrigSeries, 65
  - NIntegrate, 20
  - None, 30
  - Normal, 42
  - NormalDistribution, 56
  - NProduct, 19
  - NRoots, 19
  - NSolve, 19
  - NSum, 18
- O**
- Options, 30
- P**
- Pade, 62
  - ParametricPlot, 34
  - ParametricPlot3D, 37

PDF, 57  
PercentileBarChart, 74  
Pi, 12  
PiScale, 76  
Plot, 21, 29  
Plot3D, 29, 37  
PlotJoined, 33  
PlotLabel, 31  
PointSize, 32  
PoissonDistribution, 57  
PolarizationSum, 80  
PolarizationVector, 80  
PolarPlot, 71  
PolynomialGCD, 43  
PowerExpand, 43  
Prepend, 26  
Prime, 28  
Product, 18, 59  
PropagatorDenominator, 83

## Q

QGV, 90  
QuarkGluonVertex, 90  
QuarkSpinor, 82, 83

## R

Random, 15, 58  
Rational, 11  
Rationalize, 13  
Re, 14  
Real, 11  
Rest, 23

RGBColor, 31  
RootMeanSquare, 55  
Round, 14  
RowJoin, 53

## S

SampleRange, 55  
ScalarProduct, 79, 80, 89  
ScalarTripleProduct, 68  
Series, 42  
SetCoordinates, 68  
SetMandelstam, 87  
Show, 33  
Sign, 14  
Simplify, 45  
Sin, 16  
Solve, 45  
SP, 89  
Spherical, 68  
SphericalPlot3D, 71  
Spinor, 82, 83  
StackedBarChart, 74  
StandardDeviation, 55  
StandardDeviationMLE, 55  
StandardErrorOfSampleMean,  
55  
StandartDeviation, 58  
StudentTDistribution, 56  
SU3Delta, 82  
SU3F, 82  
Sum, 17, 40, 59  
SUND, 90

SUNDelta, 90  
SUNDeltaContract[expr], 90  
SUNF, 90  
SUNIndex, 90  
SUNN, 90  
SUNSimplify, 90  
SUNT, 90  
SUNTrace[expr], 90  
Symbol, 11  
SymbolicSum, 59

## T

Table, 25, 40  
Take, 24, 44  
TeXForm, 49  
TextListPlot, 76  
Ticks, 31  
Timing, 9  
Together, 44  
Tr, 82, 86  
Transpose, 40  
TrickMandelstam, 87  
TrigExpand, 60  
TrigFactor, 60  
TrigReduce, 60  
TrigToComplex, 60

## U

UniformDistribution, 56  
UnitStep, 63  
Unprotect, 16

## V

Variance, 55, 58  
VarianceMLE, 55  
VarianceOfSampleMean, 55

## W

While, 48

## X

XSpacing, 76

## Y

YSpacing, 76

## П

Паде-аппроксимация, 62

### а

амплитуда массива, 55  
аппроксимация  
зависимостью  $\exp(a+bx)$ ,  
28  
полиномом, 28

### в

вектор, 38  
поляризации, 80  
сопряженный, 80  
вершина в КХД  
дух-глюонная, 89  
кварк-глюонная, 90  
трехглюонная, 90

### г

генератор группы  $SU(N)$ , 90



- гистограмма, 72  
     сложенная, 74  
     трехмерная, 75  
 градиент, 69
- д**
- детерминант матрицы, 9, 40  
 дивергенция, 69  
 дисперсия  
     массива, 55  
     с максимально вероят-  
     ной оценкой, 55  
     распределения, 58  
     среднего значения  $x$ , 55  
 дифференцирование, 41
- и**
- интегрирование, 9  
     символьное, 41  
     численное, 20
- к**
- комплексно-сопряженное чис-  
 ло, 14
- л**
- лапласиан, 70
- м**
- максимум, 14, 24  
 массив, 22, 27, 54  
 матриц упорядочивание, 85  
 матрица, 26, 39  
     Гелл-Манна, 82
- Дирака, 81  
 обратная, 39  
 транспонированная, 40
- медиана, 55  
 метод  
     численного интегрирова-  
     ния, 20  
 минимум, 14, 24  
     локальный, 20  
 мнимая единица, 12  
 модуль, 9, 14
- н**
- наибольший общий делитель,  
 9
- о**
- область распределения, 57  
 оператор  
     перехода, 48  
     условный, 48  
     цикла, 48
- п**
- переменные Мандельстама,  
 87  
 плотность вероятности рас-  
 пределения, 57  
     кумулятивная, 57  
 предел, 43  
 преобразование  
     Лапласа, 66  
     обратное, 66

Фурье, 63  
    обратное, 63  
произведение, 18, 19  
    векторное, 67  
    скалярное, 39, 67  
    4-векторов, 80, 84  
    смешанное, 68  
производная, 9  
пропагаторы, 83

## Р

разложение в ряд, 42  
распределение  
     $\chi$ , 56  
     $\chi^2$ , 56  
    Бернулли, 56  
    Коши, 56  
    Лапласа, 56  
    Пуассона, 57  
    Стьюдента, 56  
    бета, 56  
    биномиальное, 56  
        отрицательное, 57  
    гамма, 56  
    геометрическое, 57  
    логарифмическое, 57  
    нормальное, 56  
    однородное, 56  
        дискретное, 57  
    экспоненциальное, 56  
решение системы уравнений  
    дифференциальных

    символьное, 45  
    численное, 21  
символьное, 45  
численное, 20  
решение уравнения  
    дифференциального  
        символьное, 45  
        численное, 21  
символьное, 45  
численное, 19, 20  
ротор, 70

## с

свертка, 84, 85, 135  
символ Кронеккера, 82, 90  
след  
    матрицы, 40  
    произведения матриц, 82  
    Гелл-Манна, 88  
    Дирака, 86  
    цветовых, 90  
случайное число, 15  
    распределения, 58  
собственные  
    вектора, 39, 40  
    значения, 9, 39, 40  
список, 54  
    с полосами ошибки, 78  
среднее  
    арифметическое  
        массива, 54  
        распределения, 57

геометрическое, 54  
среднеквадратичный корень,  
55  
стандартное отклонение  
массива, 55  
от среднего значения  $x$ ,  
55  
с максимальной вероят-  
ностной оценкой, 55  
распределения, 58  
структурная константа, 82,  
90  
сумма, 17, 18  
векторов, 39  
матриц, 39  
накопленная, 53  
символьная, 59  
суммирование по поляриза-  
циям, 80

### т

тензор  
Леви-Чевиты, 79  
метрический  $g_{\mu\nu}$ , 79, 80

### ф

факториал, 10, 16  
формат  
Mathematica, 49  
Fortran, 49  
PlainTeX, 49  
C, 49

### функция

Хевисайда, 62, 63  
многомерная, 63  
аппроксимирующая, 28

### ц

центральный момент, 54

### ч

частота, 53

### э

### элемент

матрицы, 40  
списка, 22, 54

## СВЕДЕНИЯ ОБ АВТОРЕ

Салеев Владимир Анатольевич - доктор физико-математических наук, профессор кафедры общей и теоретической физики Самарского государственного университета, заведующий кафедрой физики Самарского муниципального университета Назимовой.

Область научных интересов В.А.Салеева - квантовая теория поля, физика высоких энергий, компьютерные методы физики. Им опубликовано более 80 научных работ в ведущих научных российских и зарубежных журналах, таких как "Ядерная физика", "Physics Letters", "Modern Physics Letters", "Physical Review" и др. Научные исследования В.А.Салеева поддерживались грантами РФФИ, Министерства образования России, Президентской программы поддержки научных школ. В.А.Салеев член редколлегии журнала "Вестник Самарского университета". Он является автором ряда учебных пособий для студентов физиков: "Квантовая электродинамика лептонов и адронов. Расчеты в системе REDUCE" (1993), "Моделирование прохождения нейтронов и электронов через вещество методом Монте-Карло" (1993).