

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«САМАРСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
УНИВЕРСИТЕТ ИМЕНИ АКАДЕМИКА С.П. КОРОЛЕВА»
(САМАРСКИЙ УНИВЕРСИТЕТ)

А. В. ШИПИЛОВА

СИСТЕМА MATHEMATICA ДЛЯ МАТЕМАТИЧЕСКОГО МОДЕЛИРОВАНИЯ В ФИЗИКЕ

Рекомендовано редакционно-издательским советом федерального государственного автономного образовательного учреждения высшего образования «Самарский национальный исследовательский университет имени академика С.П. Королева» в качестве учебного пособия для обучающихся по основным образовательным программам высшего образования по направлениям подготовки 03.03.02, 03.04.02 Физика

САМАРА

Издательство Самарского университета

2026

УДК 004.94(075)+53(075)
ББК 3973.3я7+В3я7
Ш632

Рецензенты: канд. физ.-мат. наук М. В. М а р т ы н о в,
канд. физ.-мат. наук, доц. Е. В. Р о г а ч е в а

Шуилова, Александра Викторовна

Ш632 Система Mathematica для математического моделирования в физике: учебное пособие / *А. В. Шуилова*. – Самара: Издательство Самарского университета, 2026. – 124 с.: ил.

ISBN 978-5-7883-2288-9

В данном пособии представлено практическое введение в работу системы Wolfram Mathematica для математического моделирования в физике. Описан синтаксис встроенного языка Wolfram Language и показано его использование на примерах решения физических задач.

Пособие предназначено для обучающихся по направлению подготовки бакалавриата 03.03.02 Физика, магистратуры 03.04.02 Физика, студентов и аспирантов физических специальностей университетов при изучении курсов «Численные методы и математическое моделирование» и др.

Подготовлено на кафедре общей и теоретической физики.

УДК 004.94(075)+53(075)
ББК 3973.3я7+В3я7

ISBN 978-5-7883-2288-9

© Самарский университет, 2026

Оглавление

Введение	6
1 Интерфейс Wolfram Mathematica	8
1.1 Работа с системой	11
1.2 Оболочка системы и имена	14
1.3 Ввод в свободной форме	17
1.4 Скобки и группирование выражений	18
1.5 Справочная система	19
1.6 Рекомендации	22
2 Элементарные объекты	21
2.1 Числа	21
2.1.1 Целые числа	21
2.1.2 Рациональные числа	23
2.1.3 Вещественные числа	23
2.1.4 Комплексные числа	27
2.2 Бесконечность и неопределенность	29
2.3 Символы .	29
2.4 Текстовые строки	32
3 Элементарные численные выражения	33
4 Псевдослучайные числа и выборки	34
5 Функции	38
5.1 Подстановки и упрощение выражений	42
5.2 Основные группы функций	45

6	Списки	47
6.1	Генерация списков	47
6.2	Выявление структуры списков	52
6.3	Извлечение элементов списков	53
6.4	Изменение содержимого списков	55
6.5	Изменение структуры списков	56
6.6	Векторы	60
6.7	Матрицы	64
6.8	Массивы	68
7	Последовательности, суммы и произведения	70
7.1	Последовательности, ряды и пределы	70
7.2	Произведения	74
8	Ряды и пределы	75
9	Полиномы и рациональные функции	77
10	Алгебраические уравнения	79
11	Элементы математического анализа	86
11.1	Вычисление производных	86
11.2	Вычисление интегралов	88
11.3	Исследование функций на экстремум	92
11.4	Интегральные преобразования	92
12	Дифференциальные уравнения	95
12.1	Получение общего решения ДУ	95
12.2	Решение уравнений с начальными условиями	96
12.3	Численное решение дифференциальных уравнений	98

12.4 Собственные значения и собственные функции линейных дифференциальных операторов	101
13 Графика	104
13.1 Графическое представление и аппроксимация данных	104
13.2 Построение графиков	109
14 Импорт и экспорт	117
15 Элементы программирования	119
16 Полезные ресурсы	121
Список использованных источников	122

Введение

В последние годы широкое распространение получили специализированные пакеты численных вычислений (MatLab [1], Statistica [2]), системы для работы с текстом, содержащим формулы (Mathcad [3]), а также системы компьютерной алгебры или символьных вычислений. Системы Wolfram Mathematica [4], а также Maple [5] и Maxima [6] принадлежат к разряду универсальных компьютерных систем, в которых интегрированы возможности численных и аналитических вычислений, богатой компьютерной графики. Главным преимуществом Wolfram Mathematica, разработанной компанией Wolfram Research, является единая интегрированная платформа, которая объединяет вычислительное ядро, язык программирования, систему компьютерной алгебры, средства визуализации, базы данных и набор различных инструментов. В Wolfram Mathematica используется собственный мультипарадигменный язык программирования Wolfram Language [7], носящий символьный характер, то есть структура языка построена на символах и их преобразованиях. Ядро Wolfram Language можно использовать бесплатно для разработки и внедрения в коммерческие проекты с помощью бесплатной библиотеки Wolfram Engine. Wolfram Language поддерживает вызов кода на C, Java, Python и наоборот, вызов Wolfram Mathematica из этих языков. Характерными возможностями Mathematica для ее применения в научных исследованиях в физике являются:

1. Компьютерная алгебра (*Symbolic Computation*), которая умеет работать с формулами, уравнениями, интегралами, производными, многочленами, векторами, матрицами и т.д. в символьном виде, что позволяет получать точные аналитические решения.
2. Мощные алгоритмы для численного решения сложных задач (оптимизация, решение дифференциальных уравнений, линейная ал-

гебра) с произвольной точностью с целыми, рациональными, вещественными и комплексными числами.

3. Использование наиболее эффективного алгоритма для решения поставленной задачи благодаря системе автоматической выборки алгоритмов, что упрощает математическое моделирование любых процессов и выполнение компьютерного эксперимента.
4. Высококачественная визуализация: построение двумерных и трехмерных графиков, диаграмм и интерактивных элементов.
5. Удобный интерфейс в виде блокнота (*Notebook*), где код, текст, формулы, графика и результаты находятся вместе.
6. Большое количество поддерживаемых для импорта и экспорта форматов данных, включая изображения и аудиофайлы, и инструментов для статистической обработки численных, текстовых, логических и графических данных; создания баз данных.
7. Встроенные базы актуальных естественнонаучных данных.
8. Возможность подключения специализированных пакетов расширений, например, пакета FeynCalc [8], для вычислений в квантовой теории поля и физике элементарных частиц, а также разработки и анализа алгоритмов, программ и прикладных пакетов.
9. Доступ ко всем вычислительным возможностям Wolfram Mathematica на облачной платформе Wolfram Cloud [9].
10. Масштабная коллекция интерактивных образовательных приложений, созданных с помощью Wolfram Mathematica на открытой платформе Wolfram Demonstrations Project [10].

В предлагаемом учебном пособии приведено краткое введение для работы в системе Wolfram Mathematica и описание ее основных функций с примерами использования для решения физических задач, реализованных как в пакетной версии, так и в системе облачных вычислений Wolfram Cloud.

1 Интерфейс Wolfram Mathematica

Система Wolfram Mathematica реализована разработчиками в двух вариантах: пакетном, требующем установки на персональный компьютер или сервер, и в виде бесплатного открытого облачного сервиса Wolfram Cloud [9], доступного через авторизацию в интернет-браузере. В рамках сервиса Wolfram Cloud [9] выполнены все иллюстрации в данном учебном пособии. Пакетная версия является платной и доступна для установки на ОС Windows, Linux и MacOS, тогда как облачный сервис позволяет выбрать бесплатный план, подразумевающий некоторые ограничения использования, например, невозможность загрузки файлов в свой аккаунт и ограниченное время хранения рабочих файлов. Однако основной функционал системы остается доступным, за исключением проведения вычислений, требующих большого количества ресурсов.

Интерфейс пакетной и облачной версий Wolfram Mathematica очень похож, а в пакетной (десктопной) версии для упрощения ввода рядом с рабочим полем можно вызвать панели с математическими символами и греческими буквами, однако любой символ в Wolfram Mathematica возможно ввести с помощью клавиатуры, в том числе с использованием фиксированных комбинаций клавиш. После регистрации аккаунта на сайте <https://www.wolframcloud.com/> через электронную почту или аккаунты распространенных интернет-сервисов появляется стартовое окно пользователя, представленное на рис. 1. В левой части окна пользователь может выбрать создание нового блокнота (*Notebook*) одного из предложенных типов: блокнот со стандартным вводом; со вводом вида Wolfram Alpha [11]; пакет Wolfram Mathematica; html-страница и др. В данном пособии будет рассмотрена работа с блокнотами стандартного ввода. В правой части окна исходно отображается содержимое домашней папки пользователя. С

помощью верхней панели возможен доступ к сборнику документации (*Docs*), элементарному введению в Wolfram Language (*Intro*), а также полезным ссылкам (*Links*).

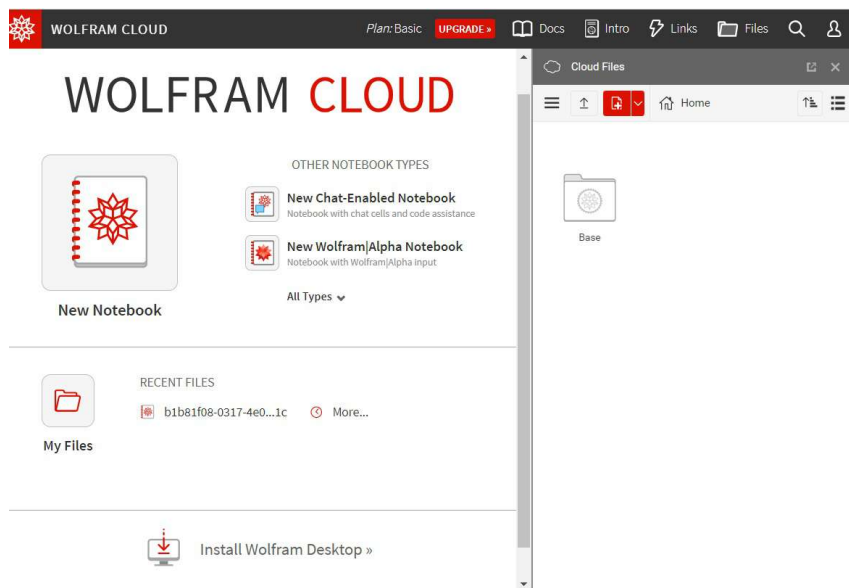


Рис. 1. Общий вид окна Wolfram Cloud

В центре документации собраны справочные материалы, содержащие подробные описания каждого элемента синтаксиса Wolfram Language с примерами использования и взаимосвязями, сгруппированные по разделам; руководства по работе с различными типами задач; связанные ресурсы, включающие в себя ссылки на вводные материалы, архивы и репозитории, ресурсы для разработчика и различные продукты компании Wolfram Research. Внешний вид всех разделов сборника документации представлен на рисунке 2.

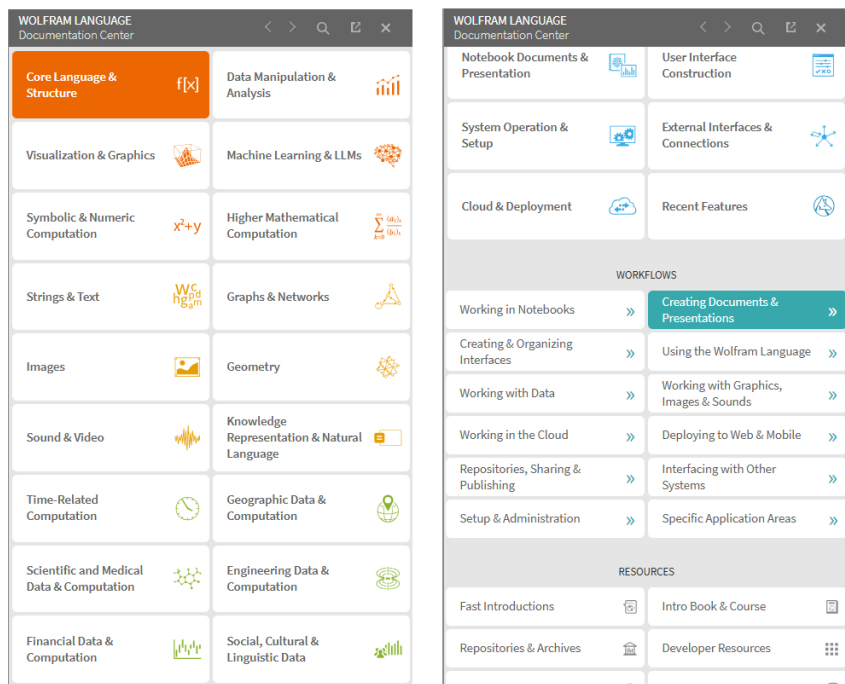


Рис. 2. Сборник документации Wolfram Cloud

После создания нового файла блокнота открывается его рабочее поле, где выполняется весь ввод и вывод. Пример простейшего вычисления в рабочем поле блокнота представлен на рис. 3:

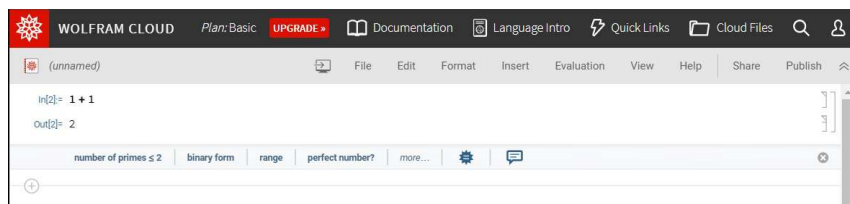


Рис. 3. Рабочее поле Wolfram Cloud

1.1 Работа с системой

При работе с системой Wolfram Mathematica основными понятиями на внутреннем уровне являются сессия (*Session*) и вычисление (*Evaluation*), на внешнем — блокнот (*Notebook*) и ячейка (*Cell*).

Сессия — период непрерывной работы ядра *MathKernel* между его вызовом и прекращением его работы по любой причине. Все функции сохраняют свои определения, а все переменные — свои значения на протяжении всей сессии, по крайней мере, до их удаления или изменения. Все введенные во время сессии, но не сохраненные в блокнот или файл определения, и все вычисленные, но не сохраненные значения теряются. С внутренней точки зрения сессия представляет собой последовательность вычислений. Для вычисления некоторого выражения нужно поместить курсор в содержащую его ячейку и нажать комбинацию клавиш *Shift + Enter*.

Любое вычисление можно приостановить либо программно, командой *Interrupt[]*, либо в диалоговом режиме главного меню командой выбором *Kernel → Interrupt Evaluation*.

Любое вычисление можно прервать программно командой *Abort[]*, либо в диалоговом режиме выбором в разделе *Kernel → Abort Evaluation*.

Первый из этих способов применяется, например, при отладке сложных программ, а второй — в случае, когда вычисление занимает неожиданно много времени, вызывая подозрение на ошибку.

Блокнот — интерактивный документ, содержащий программу, текст, результаты вычислений, сообщения, графику, таблицы и т.д. В конце сессии следует сохранить (*Save*) изменения во всех открытых блокнотах. Каждый блокнот организован как иерархическая структура, состоящая из ячеек. Каждая ячейка включает одну или несколько строк, объединенных стоящей справа скобкой. Ячейки

ввода (*Input Cell* или *Evaluatable Cell*) создаются пользователем, а ячейки вывода (*Output Cell*), содержащие сообщения об ошибках, результаты вычисления, сигналы подтверждения, статус системы и др., создаются самой системой в процессе сессии. Ячейки объединяются в группы (*Groups*). Входящие в группу ячейки соединяются общей скобкой, включающей скобки нескольких объединяемых ячеек. Например, система автоматически объединяет в одну группу ячейку ввода и получающиеся при ее исполнении ячейки вывода. Выбор пункта главного меню в десктопной версии *Cell*→*CellGrouping* позволяет произвольным образом группировать ячейки (*Group*) или разбивать имеющиеся группы (*Ungroup*).

Для создания новой ячейки нужно переместить курсор в такую позицию, где он становится горизонтальным, т.к. внутри существующей ячейки курсор всегда вертикален. Нажатие левой кнопки мыши на этой позиции курсора создает горизонтальную черту, последующее начало ввода с клавиатуры создает новую ячейку. По умолчанию она всегда имеет формат ячейки ввода, для изменения ее формата используется пункт меню *Format*. Ячейка может занимать одну или несколько строк, и всегда выделена своей квадратной скобкой справа. На ячейки накладываются ограничения: в каждой ячейке используется только один тип шрифта, нельзя комбинировать различные виды информации (например, формулы и поясняющие надписи) и т.д. При любом изменении текста формул во входной ячейке содержимое ранее выведенной выходной ячейки становится неактуальным.

Для вычисления (*Evaluate*) содержимого ячейки ввода нужно поместить курсор в эту ячейку и одновременно нажать клавиши *Shift* и *Enter*, и программа Wolfram Mathematica выдаст результат. Каждой входной/выходной строке или обращению присваивается свой номер. N -й запрос пользователя обозначается $In[N]$, а

N -й ответ начинается с $Out[N]$. Исходные данные и результаты вычислений присваиваются соответствующим объектам с использованием различных операций присваивания. Вычисляемое выражение присваивается объекту $In[N]$ с помощью операции $:=$, а результат вычисления — объекту $Out[N]$ с помощью операции $=$. Знаки $:=$ и $=$ имеют существенно различный смысл и называются отложенным и немедленным присваиванием. При немедленном присваивании выражение вычисляется сразу, а при отложенном вычисление сразу не происходит, присваивается аналитическое выражение, которое вычисляется только в момент использования. Соответственно, имеется возможность обращения к более ранним вычислениям. Для этого необходимо указать метку строки, к которой надо вернуться, например, $In[1]$, и одновременно нажать *Shift* и *Enter*. Если после входной строки не стоит никаких знаков пунктуации, то на экране появится выходная строка; в случае, когда после обращения пользователя стоит точка с запятой, выходная строка на экран не выводится, хотя подразумевается, что команда пользователя выполнена. Примеры простейшего присваивания, вычисления и выходных результатов продемонстрированы на рис. 4:

```
In[1]:=  a = 16
Out[1]=  16

In[2]:=  b = 1;
In[3]:=  c = In[1] + b
Out[3]=  17
```

Рис. 4. Пример простейшего вычисления в Wolfram Mathematica

1.2 Оболочка системы и имена

Приведем краткие правила синтаксиса Wolfram Language.

1. При наборе выражений различаются заглавные и прописные буквы. Имена всех функций, переменных, опций, констант и прочих внутренних объектов, используемых в системе Wolfram Mathematica, начинаются с заглавной буквы. Имя функции указывает цель выполняемой операции, например: *Eigenvalues* — нахождение собственных значений; *FindRoot* — численное решение уравнения; *Integrate* — интегрирование; *Timing* — выводит время, затраченное на расчет выражения. Если имя недостаточно информативно, можно воспользоваться встроенной интерактивной помощью, являющейся частью справочной базы данных системы Wolfram Mathematica. Все имена объектов, сформированных в данной программе — законченные слова, Wolfram Mathematica редко использует сокращения, за исключением общеизвестных. Некоторые из сокращений приведены в табл. 1:

Таблица 1. Сокращения в Wolfram Mathematica

<i>Abs</i>	Абсолютное значение числа
<i>Cos</i>	Значение тригонометической функции косинус
<i>D</i>	Вычисление производной
<i>Det</i>	Детерминант матрицы
<i>GCD</i>	Вычисление наибольшего общего делителя

2. Имена встроенных глобальных переменных обычно начинаются со знака *\$*, например, *\$Display*, *\$Context* и т.д. Не имеется никакого ограничения на длину имени.
3. Имена объектов, определяемых пользователем, могут начинаться с любой строчной буквы и содержать в своем составе заглавные и строчные буквы и цифры. При желании пользователь может

определять и имена, начинающиеся с заглавной буквы, но в этом случае возникает вероятность конфликта с именами встроенных объектов Wolfram Mathematica.

4. Ни одно имя не может начинаться с цифры.
5. Имя объекта не может содержать никаких специальных знаков, таких как @, %, ^, &, #, / и т.д. Эти знаки используются системой в служебных целях: для сокращенной записи функций, управления и форматирования ввода.
6. Аргументы функций всегда пишутся в квадратных скобках, например, $\cos(x)$ обозначается $\text{Cos}[x]$.
7. Фигурные скобки используются для обозначения списков, наборов и множеств.
8. Произведение обозначается как *. Пробел тоже интерпретируется как знак умножения, т. е. $x\ y$ эквивалентно $x * y$. Для понимания трактовки ввода системой следует вызвать команду просмотра полной формы вводимого выражения, см. рис. 5:

```
In[1]:= FullForm[x y]
Out[1]//FullForm=
Times[x, y]

In[2]:= FullForm[x * y]
Out[2]//FullForm=
Times[x, y]
```

Рис. 5. Вывод полной формы операции умножения

9. Возведение в степень записывается как x^y , см. рис. 6:

```
In[1]:= FullForm[x ^ y]
Out[1]//FullForm=
Power[x, y]
```

Рис. 6. Вывод полной формы операции возведения в степень

10. По умолчанию одна ячейка содержит одно вычисляемое выраже-

ние. Все пробелы, табулирования и переносы игнорируются. Однократное нажатие клавиши *Enter* начинает новую строку в текущей ячейке. Чтобы вычислить два выражения по отдельности их следует поместить в разные ячейки. Для создания новой ячейки нужно перевести курсор на одну позицию вниз посредством стрелки вниз и начать новый ввод.

11. Если для получения одного конечного выражения требуется поместить два или несколько вычисляемых выражений в одну ячейку, следует разделять их точкой с запятой, как показано на рис. 7:

```
In[1]:= 2 ^ 3 ; 3 ^ 2
```

```
Out[1]= 9
```

Рис. 7. Несколько выражений в одной ячейке с подавлением вывода

В данном случае отображается только результат последнего вычисления, т.е. знак ; подавляет вывод на экран. Если нужно увидеть оба результата, необходимо оформлять вычисление как список, в этом случае все вычисляемые выражения заключаются в фигурные скобки и разделяются запятой. Запись нового вычисления с новой строки в одной ячейке также позволяет увидеть оба результата, как на рис. 8:

```
In[2]:= {2 ^ 3 , 3 ^ 2}
```

```
Out[2]= {8, 9}
```

```
In[3]:= 2 ^ 3
```

```
3 ^ 2
```

```
Out[3]= 8
```

```
Out[4]= 9
```

Рис. 8. Несколько выражений в одной ячейке без подавления вывода

1.3 Ввод в свободной форме

Ввод знака равенства в ячейке ввода первым символом означает полное вычисление с использованием текстово-языковой формы ввода. В качестве примера выведем на экран график параболы, как показано на рис. 9.

In[1]:=



Plot a parabola

Out[1]=



Рис. 9. Пример ввода в свободной форме

Wolfram Mathematica также позволяет подобрать требуемую

функцию для выполнения задания по запросу в текстовой форме. Для этого нужно одновременно нажать комбинацию *Ctrl* =, затем в появившемся прямоугольнике записать фразу, например, *plot a sine curve*, после чего нажать *Enter*. Система выведет возможные интерпретации задания, среди которых следует принять выбранное с помощью галочки и вставить желаемый аргумент, см. рис. 10:

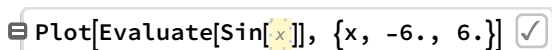


Рис. 10. Пример обработки запроса в текстовой форме

Результат запуска данного кода представлен на рис. 11.

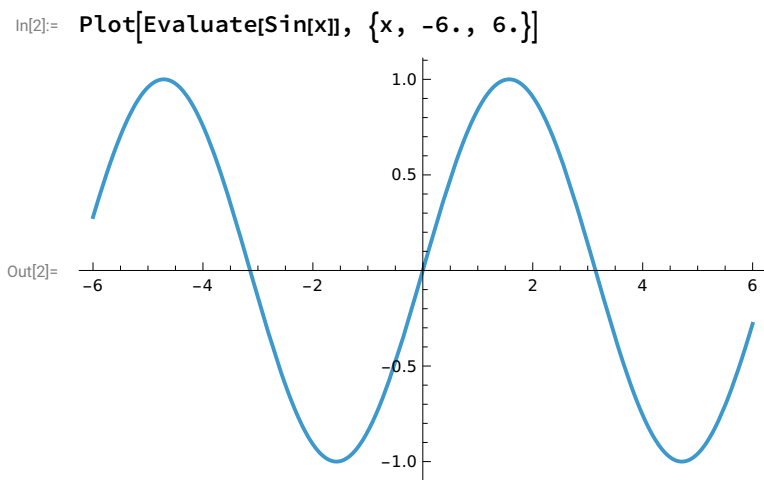


Рис. 11. Запуск кода после обработки запроса в текстовой форме

1.4 Скобки и группирование выражений

Использование различных типов скобок в системе Wolfram Mathematica жестко регламентировано. Круглые скобки () используются для группировки выражений и изменения порядка вычисления выражений, например, $(a + b)/(c + d)$.

Квадратные скобки [] являются основным признаком функций и используются при записи функций, внутри которых записываются аргументы, разделенные запятыми: $\text{Sin}[2 * \text{Pi} * x]$.

Фигурные скобки { } используются при работе с массивами, списками и матрицами: {x, y, z} — определение вектора; $\{\{a_1, b_1\}, a_2, b_2\}$ — определение матрицы.

Комментарии записываются в окружении (* *): (**текст**).

1.5 Справочная система

Все функции языка Wolfram Language документированы. Для каждой функции предоставляется: синтаксис и варианты использования, детальное описание и назначение, примеры с реальным кодом, доступным для копирования в свой блокнот, возможные опции и их значения, связанные функции и контекст. Существует несколько способов обращения к справочной системе Wolfram Mathematica.

1. При наборе пользователем нескольких символов в ячейке ввода открывается выпадающий список встроенных функций, начинающихся с данной последовательности букв. Справа от имени каждой функции стоит буква *i*, нажатие на которую открывает в правой части экрана описание соответствующей функции, полный набор ее аргументов, варианты записи, примеры использования и т.д. Этот режим интерактивной помощи включен по умолчанию и регулируется выбором в главном меню *Help* → *Hide/Show Code Captions* (скрыть/показать подписи).
2. Выбор в главном меню *Help* → *Documentation* либо *Docs* на верхней панели Wolfram Cloud открывает Центр документации Wolfram, где информация о функциях разбита по большим тематическим разделам, каждому из которых соответствует легкоузнаваемая иконка. После нажатия на иконку следует выбрать из выпадающего списка нужный подпункт, получив доступ к описанию

функций в подпункте. Представленный текст снабжен перекрестными ссылками, указаниями на схожие/родственные команды и смежные разделы. Информация по одной и той же команде может быть доступна из нескольких разных разделов.

3. Выбор в главном меню *Help* → *Language Introduction* либо *Intro* → *Elementary Introduction Book* на верхней панели Wolfram Cloud предоставляет доступ к элементарному введению в работу с Wolfram Mathematica.
4. Запись в ячейке ввода специального оператора, начинающегося со знака вопроса: *?var* — показывает общую информацию об указанной переменной величине *var*; *??var* — показывает полную информацию, включающую опции, примеры использования и др.

1.6 Рекомендации

Приведем наиболее общие рекомендации, позволяющие избежать часто возникающих ошибок и сокращающие время расчета.

1. В сложных вычислениях использовать полную форму записи выражений и задавать имена переменных обычным текстом.
2. Упрощать промежуточные результаты с помощью функций *Simplify*, *FullSimplify*, *Refine*.
3. По возможности заменять циклы работой со списками или встроенными итеративными процедурами: *Do*, *Sum*, *Product* и т.д.
4. Не применять приближенные вычисления к задачам с точными условиями. Не производить никаких округлений в процессе вычисления. Все вычисления проводить с бесконечной точностью и переходить к численным значениям только на последнем шаге.
5. Приближенные вычисления по рекуррентным формулам и в итеративных процедурах недопустимы. Выполненные без контроля сходимости и устойчивости итеративные процедуры с высокой вероятностью могут привести к ошибочному результату.

2 Элементарные объекты

Все объекты, с которыми работает Wolfram Mathematica, являются выражениями (*Expressions*) и делятся на два класса: элементарные объекты, не имеющие внутренней структуры, и составные выражения. Существует три типа элементарных объектов: числа (*Numbers*), символы (*Symbol*) и текстовые строки (*String*).

2.1 Числа

Wolfram Mathematica может производить точные операции с целыми, рациональными, вещественными и комплексными числами. Особенностью системы Wolfram Mathematica по сравнению с обычными системами численных вычислений является использование вычислений бесконечной точности. Фактически разрядность вычислений лимитируется только объемом памяти. Это, в частности, значит, что никаких округлений при вычислениях с точными вещественными числами, такими, как $\sqrt{2}$, e и π , не производится. Максимальное и минимальное числа, для которых возможно представление в данной компьютерной системе, выводятся как показано на рис. 12:

```
In[1]:= $MaxNumber
Out[1]= 1.605216761933662 × 101 355 718 576 299 609

In[2]:= $MinNumber
Out[2]= 6.229688249675322 × 10-1 355 718 576 299 610
```

Рис. 12. Максимальное и минимальное числа

2.1.1 Целые числа

В табл. 2 приведем краткий список наиболее часто используемых функций при работе с целыми числами.

Таблица 2. Функции для работы с целыми числами

<i>IntegerQ</i> [x]	Проверяет, имеет ли число x формат целого числа
<i>EvenQ</i> [n]	Проверяет n на четность
<i>OddQ</i> [n]	Проверяет n на нечетность
<i>Mod</i> [n, m]	Выводит остаток от деления n на m
<i>Quotient</i> [n, m]	Выводит неполное частное при делении n на m
<i>GCD</i> [n, m]	Выводит наибольший общий делитель l, m, n
<i>LCM</i> [l, m, n]	Выводит наименьшее общее кратное l, m, n
<i>Divisors</i> [n]	Выводит список делителей n
<i>FactorInteger</i> [n]	Раскладывает n на простые множители

Пример 1. Решим задачу, к какому типу ядер по числу протонов и нейтронов относится изотоп цезия-137. Данные об изотопах содержатся в разделе «Научные и медицинские данные & вычисления» базы данных Wolfram Mathematica и извлекаются с помощью вызова *IsotopeData*["Name", "Property"]. Перед вычислениями проверим правильность данных о массовом числе цезия-137 в базе данных и его целочисленного формата вывода, см. рис. 13.

```
In[1]:= ACs = IsotopeData["Cesium137", "MassNumber"]
Out[1]= 137

In[2]:= IntegerQ[ACs]
Out[2]= True

In[3]:= EvenQ[IsotopeData["Cesium137", "AtomicNumber"]]
Out[3]= False

In[4]:= OddQ[IsotopeData["Cesium137", "NeutronNumber"]]
Out[4]= False
```

Рис. 13. Решение задачи о четности ядра изотопа цезия-137

Следовательно, ядро цезия-137 является нечетно-четным.

2.1.2 Рациональные числа

Рациональное число x представляется как частное двух целых чисел m/n , при этом автоматически производятся все сокращения, которые система в состоянии найти, а знак переносится в числитель. Для выделения числителя (знаменателя) рационального числа x используются команды $Numerator[x]$ ($Denominator[x]$).

Пример 2. Найдем спин ядра урана-235. В случае дробного спина выделим его числитель и знаменатель, см. рис. 14.

```
In[1]:= spinU = IsotopeData["Uranium235", "Spin"]
Out[1]=  $\frac{7}{2}$ 

In[2]:= Numerator[spinU]
Out[2]= 7

In[3]:= Denominator[spinU]
Out[3]= 2
```

Рис. 14. Код для определения спина ядра изотопа урана-235

2.1.3 Вещественные числа

Вещественные и комплексные числа в Wolfram Mathematica могут быть как точными, так и приближенными. Рациональное число всегда точное, а точные иррациональные числа бывают алгебраическими и трансцендентными. Вычисления с алгебраическими числами производятся по специальным довольно сложным алгоритмам. С другой стороны, трансцендентные числа, между которыми нет очевидных зависимостей, рассматриваются как независимые полиномиальные переменные. Добавление десятичной точки к целому числу, а также к числителю и/или знаменателю рационального числа превращает их в приближенные вещественные числа.

Пример 3. Вычислим приближенное значение постоянной тон-

кой структуры $\alpha \approx \frac{1}{137}$ двумя эквивалентными способами: добавлением десятичной точки к знаменателю и с помощью функции $N[x]$, которая возвращает приближенное значение x с машинной точностью. Сравним полученные результаты с табличным значением, доступным через записи (*Entities*) базы данных Mathematica, пользуясь функцией $N[x, n]$, позволяющей получить десятичное приближение числа x с точностью до n значащих цифр, см. рис. 15.

```
In[1]:= 1/137.
Out[1]= 0.00729927

In[2]:= N[1/137]
Out[2]= 0.00729927

In[3]:= alpha = Entity["PhysicalConstant", "FineStructureConstant"]["Value"]
Out[3]= 0.00729735256

In[4]:= N[alpha, 6]
Out[4]= 0.00729735
```

Рис. 15. Вычисление значения постоянной тонкой структуры α

В обоих случаях мы получили приближенные значения α , которые совпадают с табличным с точностью до 3 значащих цифр.

Вещественное число можно представить в виде рациональной дроби командой *Rationalize*[x, dx] как на рис. 16.

```
In[5]:= Rationalize[9.6]
Out[5]=  $\frac{48}{5}$ 

In[6]:= Rationalize[0.913043]
Out[6]= 0.913043

In[7]:= Rationalize[0.913043, 0.000001]
Out[7]=  $\frac{21}{23}$ 
```

Рис. 16. Пример работы с командой *Rationalize*

Для записи числа x в стандартном и инженерном формате используются функции *ScientificForm*[x] и *EngineeringForm*[x], соответственно. В стандартной записи числа представляются в виде $a \times 10^b$, где a — число с плавающей запятой, а b — любое целое. В инженерном формате $a \times 10^{3k}$, где k — любое целое, для удобства использования кратных единиц измерения физических величин (кило, мега, гига и т.д.) Пример записи числа x в данных форматах представлен на рис. 17.

```
In[8]:= EngineeringForm[123 450.000]
Out[8]/EngineeringForm=
      123.45 × 103

In[9]:= ScientificForm[123 450.000]
Out[9]/ScientificForm=
      1.2345 × 105
```

Рис. 17. Пример записи числа в стандартном и инженерном видах

Часто используемые команды для операций с вещественными числами приведены в табл. 3.

Таблица 3. Команды для операций с вещественными числами

<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 при $x > 0$; -1 при $x < 0$
<i>Abs</i> [x]	Модуль числа x
<i>Max</i> [$x_1, x_2, \dots, x_n$]	Максимальное число из списка
<i>Min</i> [$x_1, x_2, \dots, x_n$]	Минимальное число из списка

Пример 4. Вызовем из базы данных Wolfram Mathematica атомную массу изотопа водорода ${}^1_1\text{H}$, выраженную в атомных единицах массы, и получим его массовое число с помощью кода на рис. 18.

```

In[1]:= massH = IsotopeData["Hydrogen1", "AtomicMass"]
Out[1]= 1.0078250319 u

In[2]:= Round[massH]
Out[2]= 1 u

In[3]:= IsotopeData["Hydrogen1", "MassNumber"]
Out[3]= 1

```

Рис. 18. Определение массового числа изотопа водорода 1_1H

Пример 5. Найдем максимальную энергию в системе центра масс сталкивающихся частиц, достигнутую в первом запуске (Run I) коллайдера Тэватрон (Tevatron), и радиус Большого Адронного Коллайдера (LHC) в ЦЕРНе, выраженный в метрах. Воспользуемся сведениями об ускорителях элементарных частиц (записи типа *"ParticleAccelerator"*) из базы данных Wolfram Mathematica, как показано на рис. 19.

```

In[1]:= Ceiling[Entity["ParticleAccelerator", "TevatronRunI"]["CenterOfMassEnergy"]]
Out[1]= 2 TeV

In[2]:= radiusLHC = Floor[Entity["ParticleAccelerator", "LargeHadronCollider"]["Radius"]]
Out[2]= 2 m

In[3]:= N[UnitConvert[radiusLHC], 4]
Out[3]= 3219. m

```

Рис. 19. Пример использования сведений об ускорителях

Пример 6. Найдем массы нейтрона, протона и электрона и определим максимальную и минимальную среди них. Также найдем заряды электрона и странного кварка. Воспользуемся данными из базы данных элементарных частиц и их свойств (записи типа *"Particle"*) Wolfram Mathematica как показано на рис. 20.

```

In[1]:= particleMass = {Entity["Particle", "Proton"]["Mass"],
    Entity["Particle", "Neutron"]["Mass"],
    Entity["Particle", "Electron"]["Mass"]}

Out[1]= {938.2720813 MeV/c2, 939.5654133 MeV/c2, 510.9989461 keV/c2}

In[2]:= Min[particleMass]

Out[2]= 510.9989461 keV/c2

In[3]:= Max[particleMass]

Out[3]= 939.5654133 MeV/c2

In[4]:= Sign[Entity["Particle", "Electron"]["Charge"]]

Out[4]= -1

In[5]:= Abs[Entity["Particle", "StrangeQuark"]["Charge"]]

Out[5]=  $\frac{1}{3} e$ 

```

Рис. 20. Пример работы со свойствами элементарных частиц

2.1.4 Комплексные числа

Необходимые команды для работы с комплексными числами представлены в табл. 4.

Таблица 4. Команды для работы с комплексными числами

I	Мнимая единица
$z = x + I * y$	Комплексное число z
$Re[z]$	x , вещественная часть числа z
$Im[z]$	y , мнимая часть числа z
$Conjugate[z]$	$z^* = x - iy$, комплексно-сопряженное число
$Arg[z]$	ϕ , аргумент (фаза) числа z
$Abs[z]$	$ z $, модуль числа z
$ComplexExpand[z]$	Выделение $Re(z)$ и $Im(z)$

В системе Mathematica комплексное число может быть точным или

приближенным. Комплексное число в алгебраической форме $z = x + yi$ считается точным только том случае, когда его вещественная $x = \text{Re}(z)$ и мнимая части $y = \text{Im}(z)$ являются точными.

Пример 7. Рассмотрим волновую функцию свободной одномерной квантовой частицы, воспользовавшись встроенной базой данных физических систем ("*PhysicalSystem*"). Для работы с ее действительной и мнимой частями укажем, что координате и волновому вектору соответствуют действительные числа, после чего упростим действительную и мнимую части с помощью команд *Refine* и *ComplexExpand*. Аналогичные действия продelaем с модулем и аргументом волновой функции, как показано на рис. 21.

```

In[1]:= wf = Entity["PhysicalSystem", "FreeParticle1D"][
    "ScatteringStateWaveFunctionPositionRepresentation"]
Out[1]= 
$$\frac{e^{i k x}}{\sqrt{2 \pi}}$$


In[2]:= InputForm[wf]
Out[2]/InputForm=
    "E^(I*QuantityVariable["k"], \"AngularWavenumber\")*QuantityVariable["x"], \"Length\")]/\n

In[3]:= Conditions = {Element[QuantityVariable["x", "Length"], Reals],
    Element[QuantityVariable["k", "AngularWavenumber"], Reals]}

Out[3]= {x ∈ ℝ, k ∈ ℝ}

In[4]:= Refine[Im[wf], Conditions] // ComplexExpand
Out[4]= 
$$\frac{\text{Sin}[k x]}{\sqrt{2 \pi}}$$


In[5]:= Refine[Re[wf], Conditions] // ComplexExpand
Out[5]= 
$$\frac{\text{Cos}[k x]}{\sqrt{2 \pi}}$$


In[6]:= Refine[Abs[wf], Conditions] // ComplexExpand
Out[6]= 
$$\frac{1}{\sqrt{2 \pi}}$$


In[7]:= Refine[Arg[wf], Conditions] // ComplexExpand
Out[7]= ArcTan[Cos[k x], Sin[k x]]

```

Рис. 21. Волновая функция свободной квантовой частицы

2.2 Бесконечность и неопределенность

При проведении вычислений, как арифметических, так и аналитических, в особенности при вычислении пределов, бесконечных сумм и произведений, интегралов и пр. могут возникать следующие вещественные/комплексные нечисловые объекты (*not-numbers*): *Indeterminate* — неопределенность и различные типы бесконечностей, имеющие в своем названии слово *Infinity*. Самой общей из них является функция *DirectedInfinity*. Виды бесконечностей охарактеризованы в табл. 5.

Таблица 5. Виды бесконечностей

<i>DirectedInfinity</i> [<i>z</i>]	Бесконечность в направлении комплексного числа <i>z</i>
<i>ComplexInfinity</i>	Сокращение для <i>DirectedInfinity</i> []
<i>Infinity</i>	Вещественная бесконечность, сокращение для <i>DirectedInfinity</i> [1], т.е. для направленной бесконечности вдоль положительной вещественной полуоси
<i>-Infinity</i>	Вещественная бесконечность, сокращение для <i>DirectedInfinity</i> [-1], т.е. для направленной бесконечности вдоль отрицательной вещественной полуоси

2.3 Символы

В основе любых символьных вычислений лежит способность системы работать с символами как с переменными. Эти переменные могут принимать значения любых допустимых выражений — чисел, текста или других символов — в зависимости от контекста решаемой задачи. При этом, как и в традиционном программировании, в разные моменты вычисления один и тот же символ может иметь различные

значения. То значение, которое символ имеет в данный момент вычисления, называется текущим значением. Значением символа, которому не присваивалось никаких значений, является он сам, трактуется системой как независимая переменная и печатается в блокноте синим цветом. Символы, которым присвоены значения, печатаются черным. Некоторые встроенные общеизвестные математические константы, обозначаемые защищенными символами, имеющие предопределенные численные значения, приведены в табл. 6.

Таблица 6. Математические константы

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

Встроенные математические константы в Wolfram Mathematica имеют точные значения, точность которых ограничена только компьютерной системой, на которой производятся вычисления. Приближенные значения для этих констант с точностью до n значащих цифр можно получить с помощью функции $N[name, n]$ как на рис. 22.

```
In[1]:= N[Pi, 64]
```

```
Out[1]:= 3.141592653589793238462643383279502884197169399375105820974944592
```

Рис. 22. Вывод числа π с точностью до 64 значащих цифр

Все переменные сохраняют свои значения на протяжении сессии до тех пор, пока они не будут модифицированы или очищены.

Пример 8. Рассмотрим одномерные гармонические колебания материальной точки. Присвоим переменной x выражение, соответствующее отклонению данной материальной точки, см. рис. 23.

```
In[2]:= x = x0 * Sin[w * t + phi0]
Out[2]= x0 Sin[phi0 + t w]
```

Рис. 23. Отклонение материальной точки

Укажем начальную фазу $phi0$ и выведем значение x на рис. 24.

```
In[3]:= phi0 = Pi / 2
Out[3]=  $\frac{\pi}{2}$ 

In[4]:= x
Out[4]= x0 Cos[t w]
```

Рис. 24. Значение отклонения x после присвоения значения $phi0$

Теперь вместо символа $phi0$ в значении переменной x используем его текущее значение. Изменим значение $phi0$ и вновь выведем значение x , как показано на рис. 25.

```
In[5]:= phi0 = Pi
Out[5]=  $\pi$ 

In[6]:= x
Out[6]= -x0 Sin[t w]
```

Рис. 25. Значение x после изменения значения $phi0$

Очистим значение переменной $phi0$, возвратив его к исходному символу $phi0$, и проследим за изменением значения x , см. рис. 26:

```
In[7]:= Clear[phi0]

In[8]:= x
Out[8]= x0 Sin[phi0 + t w]
```

Рис. 26. Значение x после очистки значения $phi0$

Обратим внимание на отсутствие вывода функции *Clear[phi0]* после очистки значения *phi0* и на эквивалентность знаков * и пробела при обозначении операции умножения.

2.4 Текстовые строки

Для ввода текстовой строки следует заключить текст в двойные кавычки. Для работы со строками существуют специальные функции: *StringLength* подсчитывает число символов в строке, *StringReverse* записывает символы в строке в обратном порядке, *StringJoin* объединяет строки и др.

Пример 9. Рассчитаем количество символов в слове Universe и используем его как ключевое для обращения к статье электронной энциклопедии Wikipedia. Выведем первые 105 символов данной статьи и представим их в виде отдельных строк, используя пробел как разделитель. Объединим две первых полученных строки в одну, затем запишем первую из строк в обратном порядке, см. рис. 27.

```
In[1]:= string = "Universe"
StringLength[string]
sentence = StringTake[WikipediaData[string], 105]
Out[1]= Universe
Out[2]= 8
Out[3]= The universe is all of space and time and their
        contents. It comprises all of existence, any fundamental

In[4]:= splitsentence = StringSplit[sentence]
Out[4]= {The, universe, is, all, of, space, and, time, and, their,
        contents., It, comprises, all, of, existence,, any, fundamental}

In[5]:= StringJoin[splitsentence[[1]], splitsentence[[2]]
Out[5]= Theuniverse

In[6]:= StringReverse[First[splitsentence]]
Out[6]= ehT
```

Рис. 27. Пример работы с текстовыми строками

3 Элементарные численные выражения

Все правильно составленные выражения приобретают текущие значения с учетом определений всех входящих в эти выражения функций в процессе вычисления (*Evaluation*). Например, для сложения двух чисел следует набрать первое число, затем знак $+$ и второе число, после чего нажать *Shift + Enter*. Разрядность арифметических вычислений в Mathematica лимитируется только объемом памяти. При работе с целыми числами будет выдан точный результат, даже если в нем больше цифр, чем может поместиться в одной строке, см. рис. 28.

```
In[1]:= 1234 ^ 56
Out[1]:= 129 911 902 554 871 451 941 032 084 396 235 137 754 657 820 101 273 923 843 790 127 046 242 594 330 `
550 946 489 256 784 853 624 729 020 106 139 515 647 384 910 944 921 186 523 865 849 056 275 359 066 `
262 352 911 682 504 769 929 216
```

Рис. 28. Целое число, занимающее несколько строк

При записи арифметических выражений следует учитывать приоритет арифметических операций, полные формы которых представлены в табл. 7. Например, выражение $a/b * c$ интерпретируется как $\frac{a}{b}c$. Выражение $\frac{ab}{cd}$ предпочтительно вводить в виде $(a * b)/(c * d)$.

Таблица 7. Арифметические операции в Wolfram Mathematica

Форма ввода	Внутреннее представление	Расшифровка
$x + y + z$	<i>Plus</i> [x, y, z]	Сложение
$-x$	<i>Minus</i> [x]	Переход к противоположному
$x - y$	<i>Subtract</i> [x, y]	Вычитание
$x * y * z$ или $x y z$	<i>Times</i> [x, y, z]	Умножение: знак умножения $*$ или пробел
x/y	<i>Divide</i> [x, y]	Деление
x^y	<i>Power</i> [x, y]	Возведение в степень

4 Псевдослучайные числа и выборки

Функции для генерации псевдослучайных чисел различных типов представлены в табл. 8.

Таблица 8. Функции для генерации псевдослучайных чисел

$Random[Integer]$ или $RandomInteger[]$	Генерирует псевдослучайное число, равное 0 или 1 с одинаковой вероятностью
$Random[Integer, \{I_{min}, I_{max}\}]$ или $RandomInteger[I_{min}, I_{max}]$	Генерирует целые псевдослучайные числа на отрезке $[I_{min}, I_{max}]$
$Random[]$	Генерирует вещественное псевдослучайное число на отрезке $[0,1]$
$Random[Real, \{x_{min}, x_{max}\}]$ или $RandomReal[x_{min}, x_{max}]$	Генерирует вещественное псевдослучайное число на отрезке $[x_{min}, x_{max}]$.
$Random[Complex, \{z_{min}, z_{max}\}]$ или $RandomComplex[z_{min}, z_{max}]$	Генерирует комплексное псевдослучайное число, у которого $z_{min} < z < z_{max}$

Пример 10. Выведем на экран все изотопы, у которых массовое число равно случайному числу от 1 до 3, как показано на рис. 29.

```
In[2]:= EntityClass["Isotope", "MassNumber" -> Random[Integer, {1, 3}]] // EntityList
Out[2]:= {helium-3, tritium, lithium-3}
```

Рис. 29. Изотопы со случайным массовым числом от 1 до 3

Пример 11. Выведем на экран ускорение свободного падения в случайно выбранной точке Земли следуя рис. 30.

```

In[1]:= gp = GeoPosition[{RandomReal[{-90., 90.}], RandomReal[180.]}]
Out[1]= GeoPosition[{38.3139, 7.82359}]

In[2]:= GeogravityModelData[gp]["DownComponent"]
Out[2]= 9.8092 m/s2

In[3]:= GeoNearest["City", gp]
Out[3]= {Teulada}

```

Рис. 30. Ускорение свободного падения в случайной точке

Для генерации одномерных списков псевдослучайных чисел в Wolfram Mathematica используются функции *RandomInteger*, *RandomReal*, *RandomComplex* с синтаксисом вызова: *RandomInteger*[*range*, *n*], которые генерируют список *n* соответствующих псевдослучайных чисел в интервале *range*.

Пример 12. Определим часовой пояс случайной точки в восточной части северного полушария Земли, северная широта и восточная долгота которой лежат в диапазоне от 0° до 90°, согласно рис. 31.

```

In[4]:= LocalTimeZone[GeoPosition[RandomReal[90., 2]]]
Out[4]= GMT+1

```

Рис. 31. Определение часового пояса

Пример 13. Выведем на экран первые 4 частицы с нулевым спином, присутствующие в базах данных Wolfram Mathematica, как показано на рис. 32.

```

In[1]:= RandomSample[EntityList[EntityClass["Particle", "Spin" -> 0]], 4]
Out[1]= {a0(1450)0, π(1300)+, f0(1600), f0(2100)}

```

Рис. 32. Первые 4 частицы с нулевым спином

Для генерации псевдослучайных выборок элементов заданного списка с различными весами используется команда *RandomSample* в вариантах вызова, представленных в табл. 9.

Таблица 9. Генерация псевдослучайных выборок элементов

$RandomSample[\{a_1, a_2, \dots\}, n]$	Генерирует псевдослучайную выборку n элементов из списка $\{a_1, a_2, \dots\}$
$RandomSample[\{w_1, w_2, \dots\} \rightarrow \{a_1, a_2, \dots\}, n]$	Генерирует псевдослучайную выборку n элементов из списка $\{a_1, a_2, \dots\}$ с весами $\{w_1, w_2, \dots\}$
$RandomSample[\{a_1, a_2, \dots\}]$	Генерирует псевдослучайную перестановку элементов из списка $\{a_1, a_2, \dots\}$

Генерация случайных чисел по известному закону распределения $dist$ выполняется с помощью вызова функции $RandomVariate[dist]$. Широко известные законы перечислены в табл. 10.

Таблица 10. Законы распределения случайных чисел

$NormalDistribution[\mu, \sigma]$	Гауссовское распределение $F(x) = \frac{e^{-\frac{(x-\mu)^2}{2\sigma^2}}}{\sqrt{2\pi}\sigma}$
$PoissonDistribution[\mu]$	Распределение Пуассона $F(k) = \frac{e^{-\mu}\mu^k}{k!}$
$MaxwellDistribution[\sigma]$	Распределение Максвелла $F(x) = \sqrt{\frac{2}{\pi}} \frac{x^2}{\sigma^3} \exp \frac{-x^2}{2\sigma^2}$

Пример 14. Рассчитаем среднее значение скорости $n = 10^6$ одномомерных молекул водорода при температуре $T = 293$ К, скорости которых подчиняются распределению Максвелла с параметром $\alpha = m_0/kT$, и сравним его с математическим ожиданием данного распределения с помощью кода на рис. 33.

```

In[1]:= k = Entity["PhysicalConstant", "BoltzmannConstant"]["Value"][[1]];
temp = 293; n = 10^6;
m0 = UnitConvert[MoleculeValue[Molecule["H"]], "MolecularMass"], "Kilograms"][[1]]
Out[3]= 3.348 × 10-27

In[4]:= alpha = m0 / (k * temp);
Total[RandomVariate[MaxwellDistribution[alpha], n]]/n
Out[5]= 1.32069 × 10-6

In[6]:= Mean[MaxwellDistribution[alpha]]
Out[6]= 1.321 × 10-6

```

Рис. 33. Расчет средней скорости молекул водорода

5 Функции

В Mathematica можно работать с большим количеством элементарных и специальных функций, имена которых обычно соответствуют их названиям на английском языке. Численное значение функции f при определенном значении аргумента x может быть получено командой $N[f[x]]$. Образ элемента x под действием f вводится одним из эквивалентных способов, см. рис. 34:

1. $f[x]$ — функциональная запись.
2. $f@x$ и x/f — префиксная и постфиксная операторная записи, соответственно. С их помощью удобно строить цепочки функций для быстроты набора и чтения кода.
3. $Function[z, f[z]][x]$ или $f[\#]&[x]$ — запись в виде чистой (анонимной) функции. Чистые функции позволяют применять функции к аргументам без необходимости определять явные имена для этих функций. В случае сокращенной записи через символы $\&$ и $\#$ символ $\#$ (или $\#1$) заменяется первым аргументом, $\#2$ вторым и так далее. Символ $\#0$ заменяется самой функцией.

```
In[1]:= Sin[x]
Out[1]= Sin[x]

In[2]:= Sin@x
Out[2]= Sin[x]

In[3]:= x // Sin
Out[3]= Sin[x]

In[6]:= Function[z, z^2 + z + 1][x]
Out[6]= 1 + x + x^2

In[7]:= (#1 + #2)^2 &[x, y]
Out[7]= (x + y)^2
```

Рис. 34. Способы задания функции

Постфиксная форма // имеет более низкий приоритет, чем все арифметические и логические операции. В большинстве случаев функция, примененная при помощи // будет применяться ко всему выражению в целом, что позволяет удобно пользоваться постфиксной формой для просмотра или форматирования вывода. Напротив, префиксная форма @ имеет наивысший приоритет среди всех арифметических и большинства других операций, см. рис. 35.

```
In[1]:= z = 2 + I * 2
Out[1]= 2 + 2 i

In[2]:= (Exp[z] + z) // Exp // Conjugate // ComplexExpand
Out[2]= e2+e2 Cos[2] Cos[2 + e2 Sin[2]] - i e2+e2 Cos[2] Sin[2 + e2 Sin[2]]

In[3]:= (Exp[z] + (z // Exp)) // Conjugate // ComplexExpand
Out[3]= 2 e2 Cos[2] - 2 i e2 Sin[2]

In[4]:= Exp[z] + (z // Exp // Conjugate) // ComplexExpand
Out[4]= 2 e2 Cos[2]

In[5]:= Sin@z + Conjugate[z]
Out[5]= (2 - 2 i) Sin[2 + 2 i]

In[6]:= Sin@(z * Conjugate[z])
Out[6]= Sin[8]
```

Рис. 35. Свойства префиксной и постфиксной форм записи функций

Пример 15. Выведем в секундах время расчета среднего расстояния от Солнца астрономических объектов внутренней части главного пояса астероидов, см. рис. 36:

```
In[1]:= ast = MinorPlanetData[EntityClass["MinorPlanet", "InnerMainBeltAsteroid"],
"DistanceFromSun"];
UnitConvert@Timing@Mean@ast
Out[2]= {0.009299, 2.8903 × 1011 m}
```

Рис. 36. Вывод времени расчета

Пример 16. Выведем численное значение $e^{\sqrt{i}}$, см. рис. 37.

```
In[1]:= I // Sqrt // Exp // ComplexExpand // N  
Out[1]= 1.54186 + 1.31754 i
```

Рис. 37. Вывод численного значения $e^{\sqrt{i}}$

Одна и та же функция или команда может вести себя различным образом в зависимости от типа объектов, к которым она применяется, т.к. изменяются ее опции по умолчанию и используемые алгоритмы. Она также может вызываться с разным количеством аргументов, которые имеют разный статус. Аргументы встроенных функций подразделяются на собственно аргументы, параметры, опции и атрибуты. Собственно аргументы — обычные аргументы в математическом понимании функции, все они обязаны явно фигурировать в задании функции в виде $f[x, y]$, тогда как указание параметров необязательно. Опции и атрибуты описывают режим работы функции, используемые при ее вычислении параметры, тождества и процедуры, для которых существуют стандартные настройки. Эти настройки можно изменять: опции меняются при помощи специальной конструкции *Rule*, посредством включения текста *Option* $\rightarrow$ *Choice* в тело функции, а атрибуты — внешним образом при помощи специальных команд, таких как *SetAttributes*. Например, функция *Log*, вызванная с единственным аргументом в виде вещественного числа x , выдает значение натурального логарифма этого числа. Та же функция, вызванная с двумя аргументами в полном виде $\text{Log}[b, x]$, выдаст значение логарифма по заданному основанию b , см. рис. 38.

Пользователь может определить собственную именованную функцию с помощью конструкции $f[x_]:=$, где запись $x_$ означает шаблон, вместо которого может быть подставлено любое значение. Знак отложенного присваивания $:=$ означает, что любой аргумент, переданный f , подставляется в правую часть при вычислении.

```
In[1]:= a = 100.; b = 10.
```

```
Out[1]= 10.
```

```
In[2]:= Log[a]
```

```
Out[2]= 4.60517
```

```
In[3]:= Log[b, a]
```

```
Out[3]= 2.
```

Рис. 38. Вызовы функции *Log* с различными аргументами

Пример 17. Рассчитаем энергию связи ядер ${}^4_2\text{He}$ и ${}^{235}_{92}\text{U}$ используя формулу Вайцзеккера-Вильямса

$$E = a_1 A - a_2 A^{2/3} - a_3 Z^2 / A^{1/3} - a_4 (A - 2Z)^2 / A + a_5 A^{-3/4},$$

где $a_1 = 15.75$ МэВ, $a_2 = 17.8$ МэВ, $a_3 = 0.711$ МэВ, $a_4 = 23.7$ МэВ, $a_5 = 34$ МэВ (четно-четные ядра), 0 (нечетные ядра), -34 МэВ (нечетно-нечетные ядра), как показано на рис. 39.

```
In[1]:= a1 = 15.75; a2 = 17.8; a3 = 0.711; a4 = 23.7; a5 = 34.;
```

```
In[2]:= WW[Z_, A_] :=
```

```
a1 * A - a2 * A ^ (2 / 3) - a3 * Z ^ 2 * A ^ (-1 / 3) -  
a4 * (A - 2 * Z) ^ 2 / A + a5 * A ^ (-3 / 4) * (Mod[A, 2] + Mod[Z, 2] - 1) * (-1)
```

```
In[3]:= WW[2, 4] / 4
```

```
Out[3]= 7.094
```

```
In[4]:= WW[92, 235] / 235
```

```
Out[4]= 7.59957
```

Рис. 39. Расчет энергии связи ядер ${}^4_2\text{He}$ и ${}^{235}_{92}\text{U}$

Рекурсивное n -кратное применение заданной функции f к выражению $expr$ можно выполнить с помощью функции $Nest[f, expr, n]$.

Возможно отдельно определить значение функции для данного значения аргумента.

Пример 18. Рассчитаем $n!$ с помощью кода на рис. 40.

```

In[1]:= factorial[0] = 1
Out[1]= 1

In[2]:= factorial[n_Integer] := n*factorial[n - 1]

In[3]:= factorial[10]
Out[3]= 3 628 800

```

Рис. 40. Расчет $n!$

Обратим внимание, что названия встроенных функций защищены и могут быть использованы пользователем в других целях только после снятия защиты командой *Unprotect[FunctionName]*. Поэтому строчная буква в названии функции *factorial* в примере 18 выбрана во избежание конфликта со встроенной функцией *Factorial*.

5.1 Подстановки и упрощение выражений

Применение замен и правил подстановки к выражениям производится при помощи операторов замены, таких как *Replace* и его усиленных вариантов, представленных в табл. 11.

Таблица 11. Операторы замены

<i>/.</i> <i>ReplaceAll[expr, rules]</i>	или	Однократная замена согласно списку правил <i>rules</i> в выражении <i>expr</i> на всех уровнях
<i>//.</i> <i>ReplaceRepeated[expr, rules]</i>	или	Множественная замена согласно списку правил <i>rules</i> , повторяемая до тех пор, пока выражение <i>expr</i> не прекратит изменения

Пример 19. Применим команду *ReplaceAll* для нахождения функций сферических гармоник, соответствующих электрону в атоме водорода с главным квантовым числом $n = 2$ и орбитальным квантовым числом $l = 0, 1, 2$, как показано на рис. 41.

```
In[1]:= ReplaceAll[SphericalHarmonicY[n, l,  $\theta$ ,  $\phi$ ],
{{n  $\rightarrow$  2, l  $\rightarrow$  2}, {n  $\rightarrow$  2, l  $\rightarrow$  1}, {n  $\rightarrow$  2, l  $\rightarrow$  0}}]
Out[1]=  $\left\{ \frac{1}{4} e^{2i\phi} \sqrt{\frac{15}{2\pi}} \sin[\theta]^2, -\frac{1}{2} e^{i\phi} \sqrt{\frac{15}{2\pi}} \cos[\theta] \sin[\theta], \frac{1}{4} \sqrt{\frac{5}{\pi}} (-1 + 3 \cos[\theta]^2) \right\}$ 
```

Рис. 41. Использование команды *ReplaceAll*

Пример 20. Выполним многократную замену после пятикратного рекурсивного применения функции *Sin*, как показано на рис. 42.

```
In[1]:= expr = Nest[Sin, x, 5]
Out[1]= Sin[Sin[Sin[Sin[Sin[x]]]]]

In[2]:= expr //. {Sin[x_]  $\rightarrow$  y}
Out[2]= y
```

Рис. 42. Использование команды *ReplaceRepeated*

Если в процессе отладки программы приходится часто придавать переменным значения посредством операторов присваивания, то время от времени эти переменные следует очищать при помощи специальных команд, представленных в табл. 12.

Таблица 12. Команды очистки переменных

$x = .$ или <i>Unset</i> [x]	Очистить значения x
<i>Clear</i> [x, y, z]	Очистить значения и определения x, y, z
<i>ClearAll</i> [x, y, z]	Очистить значения, определения, атрибуты и опции x, y, z
<i>Remove</i> [x, y, z]	Полностью очистить переменные x, y, z и забыть их имена

Полезными командами всей системы Wolfram Mathematica являются команды упрощения *Simplify*, *FullSimplify*, *Refine* и *Assuming*, действие которых описано в табл. 13, где в угловых скобках указан необязательный параметр — дополнительные условия, которые предполагается использовать в процессе упрощения.

Таблица 13. Команды упрощения выражений

$Simplify[expr, \langle asn \rangle]$, $FullSimplify[expr, \langle asn \rangle]$	Упрощает выражения $expr$ с помощью алгебраических и прочих преобразований
$Refine[expr, \langle asn \rangle]$	Приводит выражение $expr$ к виду, когда все входящие в него символьные выражения предполагаются численными, удовлетворяющими заданным условиям
$Assuming[asn, expr]$	Вычисляет $expr$, предполагая выполнение условий asn

Пример 21. Рассмотрим применение команд упрощения для часто возникающих выражений в квантовой физике на рис. 43.

```
In[1]:= Psi = A1 * Exp[I * k * x] + A2 * Exp[-I * k * x]
Out[1]= A2 e-i k x + A1 ei k x

In[2]:= Psi /. {A1 -> 1, A2 -> 1}
Out[2]= e-i k x + ei k x

In[3]:= FullSimplify[%]
Out[3]= 2 Cos[k x]

In[4]:= Refine[Sin[a + Pi * n], Assumptions -> {n ∈ Integers}]
Out[4]= (-1)n Sin[a]

In[5]:= expr = (energy / c + momentum) * (energy / c - momentum)
Out[5]=  $\left(\frac{\text{energy}}{c} - \text{momentum}\right) \left(\frac{\text{energy}}{c} + \text{momentum}\right)$ 

In[6]:= sqp = Sqrt[expr] /. energy -> Sqrt[(momentum ^ 2 + m ^ 2 * c ^ 2) * c ^ 2] // Simplify
Out[6]=  $\sqrt{c^2 m^2}$ 

In[7]:= Assuming[{m > 0, c > 0}, Simplify[sqp]]
Out[7]= c m
```

Рис. 43. Упрощение выражений

Основное различие между *Simplify* и *FullSimplify* состоит в ограничениях по времени на выполнение определенного преобразования. Для команды *Simplify* по умолчанию *TimeConstraint* $\rightarrow$ 300, а для *FullSimplify* по умолчанию *TimeConstraint* $\rightarrow$ *Infinity*, кроме того, она использует более широкий список табличных преобразований и не накладывает ограничений на длину промежуточных результатов. Для числовых функций рекомендуется использовать команду *Refine*.

5.2 Основные группы функций

Набор простейших функций, работающих на различных числовых множествах, приведен в разделе 2.1. Список всех функций, доступных в Wolfram Mathematica, является чрезвычайно объемным и содержит в себе практически все математические конструкции, объекты и инструменты, обычно используемые для решения задач в физике, включая специальные функции, например, дельта-функцию Дирака. Полный набор функций различных типов представлен в Центре документации Wolfram Mathematica, где они объединены в разделы, причем одна функция может иметь описание сразу в нескольких разделах, список которых приведен ниже.

1. Язык ядра и структура (Core Language & Structure).
2. Символьные и численные вычисления (Symbolic & Numeric Computation).
3. Вычисления в высшей математике (Higher Wolfram Mathematical Computation).
4. Визуализация и графика (Visualization & Graphics).
5. Обработка и анализ данных (Data Manipulation & Analysis).
6. Строки и текст (Strings & Text).
7. Сети и графы (Graphs & Networks).
8. Изображения (Images).

9. Геометрия (Geometry).
10. Звук и видео (Sound & Video).
11. Машинное обучение и нейросети (Machine Learning & LLMs).
12. Вычисления, связанные со временем (Time-Related Computation).
13. Научные и медицинские расчеты и данные (Scientific and Medical Data & Computation).
14. Географические расчеты и данные (Geographic Data & Computation).
15. Инженерные расчеты и данные (Engineering Data & Computation).
16. Социальные, культурные и лингвистические данные (Social, Cultural & Linguistic Data).
17. Финансовые расчеты и данные (Financial Data & Computation).
18. Представление знаний и естественный язык (Knowledge Representation & Natural Language).
19. Блокноты с документами и презентациями (Notebook Documents & Presentation).
20. Конструирование пользовательского интерфейса (User Interface Construction).
21. Взаимодействие с облаком и размещение данных (Cloud & Deployment).
22. Работа с системой и установка (System Operation & Setup).
23. Внешние связи и интерфейсы (External Interfaces & Connections).
24. Обновления (Recent Features).

6 Списки

С точки зрения языка системы Wolfram Mathematica и внутреннего представления данных множества, наборы, последовательности, векторы, матрицы, тензоры трактуются как списки (list). Поэтому грамотное программирование на языке Wolfram Language подразумевает объединение переменных, команд, уравнений, условий, подстановок в списки для работы с ними вместо явного использования циклов. Список, элементы которого сами являются списками, называется вложенным списком (nested list).

В Wolfram Mathematica список обозначается $\{x_1, \dots, x_n\}$ или, в полной форме, $List[x_1, \dots, x_n]$, см. рис. 44:

```
In[1]:= List[a, b, c, d]
Out[1]= {a, b, c, d}
```

Рис. 44. Задание списка

В системе содержится несколько десятков операций над списками, позволяющих генерировать списки, выявлять и изменять их структуру, извлекать элементы, изменять содержимое списков и применять функции вида *Map*, *Apply*, *Thread*, *Inner*, *Outer* и их варианты.

6.1 Генерация списков

Основным способом генерации векторов, матриц и других списков в языке Wolfram Language является функция *Table*, способы вызова которой представлены в табл. 14. Матрица в Mathematica трактуется как строка, составленная из строк, при этом внутренние итераторы пишутся последними. Вызов $Table[f[i, j], \{i, i_{min}, i_{max}\}, \{j, j_{min}, j_{max}\}]$ с двумя итераторами порождает вложенный список значений функции двух аргументов f в парах (i, j) , который организован как матрица, где итератор i считается внешним, а j — внутренним, т.е. i нумерует строки, а j — столбцы матрицы. Таким образом,

$Table[f[i, j], \{i, i_{min}, i_{max}\}, \{j, j_{min}, j_{max}\}, \dots]$ генерирует вложенный (многомерный) список, где самым внешним является список по i , а наиболее внутренние итераторы пишутся последними.

Таблица 14. Способы генерации списков

$Table[expr, n]$	Генерирует список, содержащий n экземпляров выражения $expr$
$Table[expr, \{i, i_{max}\}]$	Генерирует список значений $expr$ при i , изменяющемся от 1 до i_{max} с шагом 1
$Table[expr, \{i, i_{min}, i_{max}\}]$	Генерирует список значений $expr$ при i , изменяющемся от i_{min} до i_{max} с шагом 1
$Table[expr, \{i, i_{min}, i_{max}, di\}]$	Генерирует список значений $expr$ при i , изменяющемся от i_{min} до i_{max} с шагом di

Простейшим примером использования функции $Table$ является подготовка таблицы с начальными значениями некоторой физической величины, как показано на рис. 45:

```
In[1]:= zerolist = Table[0.0, 10]
Out[1]= {0., 0., 0., 0., 0., 0., 0., 0., 0., 0.}
```

Рис. 45. Задание простейшей таблицы

Пример 22. Занесем в таблицу изменение координаты точки, движущейся по гармоническому закону $x = 4 \cos(0.8t)$ за первые 10 секунд движения и представим полученную зависимость графически, как продемонстрировано на рис. 46.

Пример 23. Рассмотрим зависимость энергии связи от массового числа всех стабильных изотопов из базы данных Wolfram Mathematica, как показано на рис. 47.

```
In[1]:= harmonic1 =
Table[x0 * Cos[omega * t], {t, 10}] /. {x0 -> 4., omega -> 0.8}
Out[1]:= {2.78683, -0.116798, -2.94957, -3.99318,
-2.61457, 0.349996, 3.10226, 3.97274, 2.43341, -0.582}
```

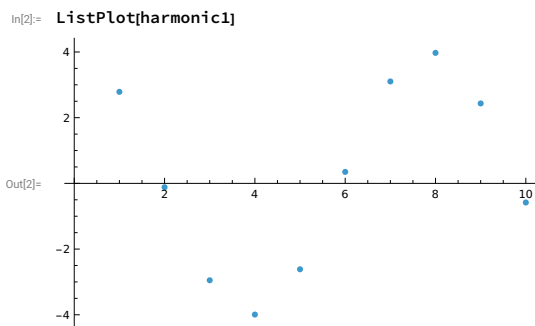


Рис. 46. Иллюстрация гармонического закона движения точки

```
In[1]:= stablelist = EntityList[EntityClass["Isotope", "Stable"]];
In[2]:= energyT =
Table[{stablelist[[i]]["MassNumber"], stablelist[[i]]["BindingEnergy"]}, {i, Length[stablelist]}];
In[3]:= ListPlot[energyT, Frame -> True, FrameLabel -> {"Массовое число", "Энергия связи"},
FrameStyle -> Thick, LabelStyle -> Directive[Bold, Medium]]
```

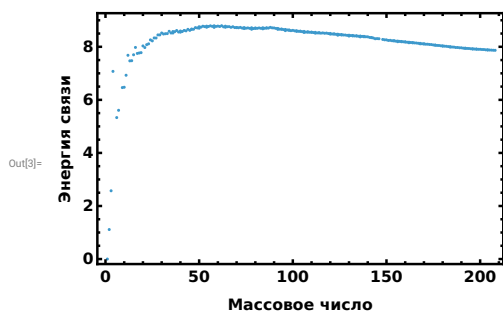


Рис. 47. Энергия связи стабильных изотопов

Пример 24. Рассмотрим спектр излучения абсолютно черного тела (АЧТ) при температуре 10000 К. Занесем в таблицу значения спектральной плотности энергетической светимости АЧТ в зависимости от длины волны в пределах $50 \text{ нм} \leq \lambda \leq 2500 \text{ нм}$ с шагом 50 нм, пользуясь встроенной в Wolfram Mathematica функцией

PlanckRaditaionLaw. Выведем полученные точки в виде графика с помощью функции *ListPlot*, как показано на рис. 48 и сохраним его как графический объект *plotPlanck*. Выведем на экран плавный график той же зависимости указав свойство *SpectralPlot* в функции *PlanckRaditaionLaw* и сохраним его как *fig1*, см. рис. 49.

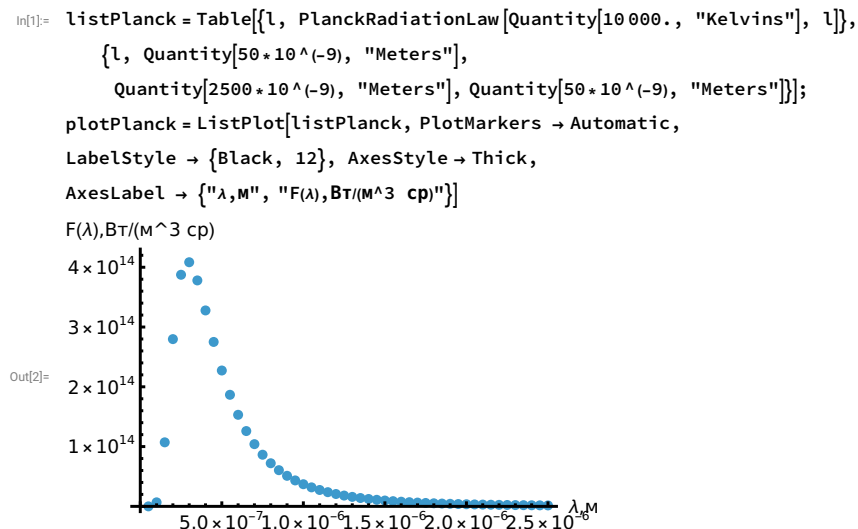


Рис. 48. Точечный график спектра излучения АЧТ при $T = 10^4$ К

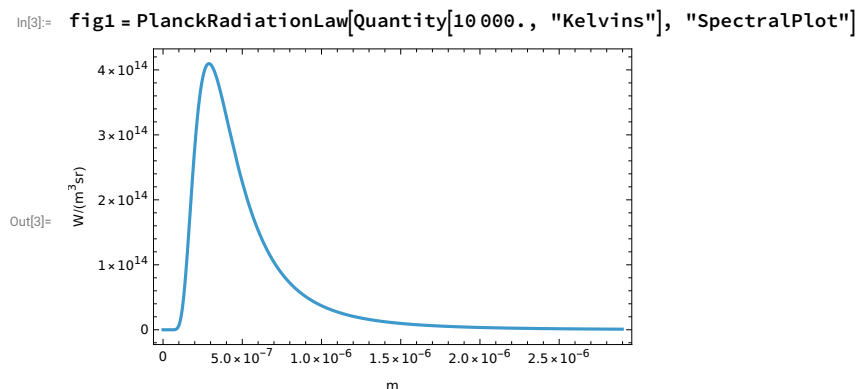


Рис. 49. Плавный график спектра излучения АЧТ при $T = 10^4$ К

Выведем на экран одновременно оба графика с помощью функции *Show* и отметим их совпадение на рис. 50.

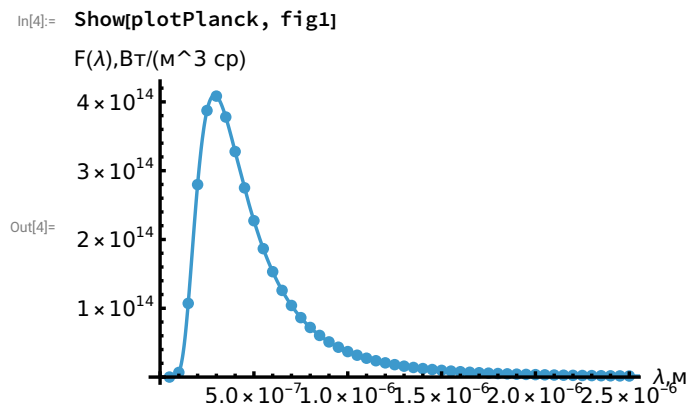


Рис. 50. Спектр излучения АЧТ при $T = 10^4 \text{ К}$

Пример 25. Зададим единичную матрицу с помощью символа Кронекера и встроенной функции *IdentityMatrix*[3] и проверим идентичность результатов, как показано на рис. 51.

```
In[1]:= id1 = Table[KroneckerDelta[i, j], {i, 3}, {j, 3}] // MatrixForm
```

Out[1]//MatrixForm=

$$\begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

```
In[2]:= id2 = IdentityMatrix[3] // MatrixForm
```

Out[2]//MatrixForm=

$$\begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

```
In[3]:= id1 == id2
```

Out[3]= True

Рис. 51. Задание единичной матрицы

6.2 Выявление структуры списков

Функции данного раздела представлены в табл. 15.

Таблица 15. Функции выявления структуры списков

<i>Length</i> [<i>list</i>]	Возвращает количество элементов в списке <i>list</i>
<i>VectorQ</i> [<i>list</i>]	Возвращает <i>True</i> , если выражение является одномерным вектором, т. е. не содержит вложенных списков, иначе <i>False</i>
<i>Count</i> [<i>list</i> , <i>pattern</i>]	Подсчитывает, сколько элементов списка <i>list</i> удовлетворяют определенному условию (шаблону) <i>pattern</i>
<i>Position</i> [<i>list</i> , <i>pattern</i>]	Возвращает номера позиций в списке <i>list</i> , на которых расположены элементы, удовлетворяющие шаблону <i>pattern</i>
<i>Total</i> [<i>list</i>]	Возвращает сумму элементов списка <i>list</i>

Пример 26. Организуем в список записи, содержащие информацию о всех изотопах йода, присутствующих в базе данных Wolfram Mathematica, выясним их количество и проверим, что упорядоченные командой *EntityList* записи действительно являются одномерным списком, составленным из записей, как показано на рис. 52.

```

In[1]:= iodclass = EntityClass["Isotope", "Iodine"];
        iodlist = EntityList[iodclass];
        Length[iodlist]
Out[3]= 42

In[4]:= VectorQ[iodlist]
Out[4]= True

In[5]:= Count[iodlist, _Entity, Infinity]
Out[5]= 42

```

Рис. 52. Список, составленный из записей

Пример 27. Выведем позицию в списке изотопа йода ^{131}I и считаем среднюю по всем изотопам йода энергию связи на рис. 53:

```
In[6]:= Position[EntityValue[iodclass, "MassNumber"], 131]
Out[6]:= {{26}}

In[7]:= N[Total[EntityValue[iodclass, "BindingEnergy"]]/
  Length[EntityList[iodclass]]]
Out[7]:= 8.31248 MeV
```

Рис. 53. Позиция элемента и среднее значение элементов списка

6.3 Извлечение элементов списков

Для выделения частей списков в языке Wolfram Mathematica используются двойные квадратные скобки `[[ ]]`, являющиеся сокращением команды *Part*. При этом номера позиций частей в списке тоже могут задаваться списком, а для выбора всех частей на каком-либо уровне нужно воспользоваться спецификацией *All*. Функции, позволяющие оперировать элементами списков, приведены в табл. 16.

Пример 28. Выведем массовые числа и периоды полураспада изотопов урана и нептуния, см. рис. 54.

```
In[1]:= uranium = IsotopeData["Uranium"];
neptunium = IsotopeData["Neptunium"];

In[3]:= IsotopeData[First[neptunium], "MassNumber"]
Out[3]:= 219

In[4]:= IsotopeData[Last[neptunium], "HalfLife"]
Out[4]:= 6. min

In[5]:= Max[{IsotopeData[uranium, "HalfLife"],
  IsotopeData[neptunium, "HalfLife"]}]]
Out[5]:= 4.46 Gyr
```

Рис. 54. Свойства изотопов урана и нептуния

Таблица 16. Функции для извлечения частей списков

<i>First</i> [<i>list</i>]	Возвращает первый элемент списка
<i>Last</i> [<i>list</i>]	Возвращает последний элемент списка
<i>Part</i> [<i>list</i> , <i>n</i>] или <i>list</i> [[<i>i</i>]]	Возвращает <i>n</i> -й элемент списка
<i>Take</i> [<i>list</i> , <i>n</i>] (<i>Take</i> [<i>list</i> , $-n$])	Возвращает <i>n</i> первых (последних) элементов списка <i>list</i>
<i>Take</i> [<i>list</i> , <i>m</i> , <i>n</i>]	Возвращает элементы списка от <i>m</i> до <i>n</i>
<i>Drop</i> [<i>list</i> , <i>m</i> , <i>n</i>]	Возвращает список без элементов от <i>m</i> до <i>n</i>
<i>Rest</i> [<i>list</i>]	Возвращает список без последнего элемента
<i>Min</i> [$x_1, x_2, \dots$] (<i>Max</i> [$x_1, x_2, \dots$])	Возвращает наименьшее (наибольшее) из числовых значений x_i
<i>Min</i> [$\{x_1, x_2, \dots\}, \{y_1, y_2, \dots\}, \dots$] (<i>Max</i> [$\{x_1, x_2, \dots\}, \{y_1, y_2, \dots\}, \dots$])	Возвращает наименьший (наибольший) элемент любого из списков $\{x_1, x_2, \dots\}, \{y_1, y_2, \dots\}$

Пример 29. Выполним ряд действий с четырьмя первыми свойствами изотопов урана-235 как представлено на рис. 55.

```

In[1]:= u235prop = IsotopeData["Uranium235"]["Properties"];
u235prop4 = Take[u235prop, 4]

Out[2]= { atomic mass , atomic number , atomic symbol , binding energy per nucleon }

In[3]:= IsotopeData["Uranium235"][u235prop4[[2]]]

Out[3]= 92

In[4]:= Drop[u235prop, {2, Length[u235prop] - 1}]

Out[4]= { atomic mass , width }

In[5]:= IsotopeData["Uranium235"][Rest[%]]

Out[5]= { 2.05 × 10-32 eV }

```

Рис. 55. Свойства изотопов урана-235

6.4 Изменение содержимого списков

Основными операциями при работе со списками является добавление и удаление элементов списков, соответствующие функции приведены в табл. 17.

Таблица 17. Функции для изменения содержимого списков

<i>Append</i> [<i>list</i> , x_i]	Добавляет элемент x_i в конец списка <i>list</i>
<i>Prepend</i> [<i>list</i> , x_i]	Добавляет элемент x_i в начало списка <i>list</i>
<i>Insert</i> [<i>list</i> , x_i , <i>n</i>]	Добавляет элемент x_i на <i>n</i> -ю позицию списка <i>list</i>
<i>ReplacePart</i> [<i>list</i> , x_i , <i>n</i>]	Возвращает список, в котором элемент, стоящий на позиции с номером <i>n</i> , заменяется на элемент x_i
<i>Delete</i> [<i>list</i> , <i>n</i>]	Возвращает список <i>list</i> , в котором удален элемент на позиции <i>n</i>
<i>DeleteCases</i> [<i>list</i> , <i>pattern</i>]	Позволяет производить удаление по шаблону

Пример 30. Рассмотрим изменение содержимого списков на примере работы с экспериментальными данными в виде значений времени колебаний математического маятника. Внесем очередное значение и найдем минимальное значение в списке, ошибочно внесенное с пропущенным нулем. Определим его позицию в списке и заменим значение на данной позиции результатом верного измерения, после чего озаглавим список. Рассчитаем среднее время колебаний, исключив из расчета первое значение списка, являющееся заголовком. Занесем в список отклонение от среднего времени колебаний в каждом измерении и представим полную таблицу, включающую в

себя экспериментальные данные, среднее значение и отклонения от среднего, как продемонстрировано на рис. 56.

```

In[1]:= TimeData = {20.1, 20.2, 20.8, 2.0, 20.2, 20.3, 20.5, 20.4}
Out[1]= {20.1, 20.2, 20.8, 2., 20.2, 20.3, 20.5, 20.4}

In[2]:= Append[TimeData, 20.3]
Out[2]= {20.1, 20.2, 20.8, 2., 20.2, 20.3, 20.5, 20.4, 20.3}

In[3]:= Min[TimeData]
Out[3]= 2.

In[4]:= PositionSmallest[TimeData]
Out[4]= {4}

In[5]:= TimeData = ReplacePart[TimeData, 20.0, 4]
Out[5]= {20.1, 20.2, 20.8, 20., 20.2, 20.3, 20.5, 20.4}

In[6]:= TimeData = Prepend[TimeData, "Время колебаний"]
Out[6]= {Время колебаний, 20.1, 20.2, 20.8, 20., 20.2, 20.3, 20.5, 20.4}

In[7]:= meanTime = Mean[DeleteCases[TimeData, _String]]
Out[7]= 20.3125

In[8]:= DeltaTime =
  ReplacePart[TimeData - meanTime, "Отклонение от среднего", 1]
Out[8]= {Отклонение от среднего, -0.2125, -0.1125,
  0.4875, -0.3125, -0.1125, -0.0125, 0.1875, 0.0875}

In[9]:= FullTime = {TimeData, DeltaTime}
Out[9]= {{Время колебаний, 20.1, 20.2, 20.8, 20., 20.2, 20.3, 20.5, 20.4}, {Отклонение от среднего,
  -0.2125, -0.1125, 0.4875, -0.3125, -0.1125, -0.0125, 0.1875, 0.0875}}

In[10]:= FullTime0 =
  Insert[FullTime,
  Prepend[Table[meanTime, Length[TimeData] - 1], "Среднее время"], 2]
Out[10]= {{Время колебаний, 20.1, 20.2, 20.8, 20., 20.2, 20.3, 20.5, 20.4},
  {Среднее время, 20.3125, 20.3125, 20.3125, 20.3125, 20.3125, 20.3125, 20.3125, 20.3125},
  {Отклонение от среднего, -0.2125, -0.1125,
  0.4875, -0.3125, -0.1125, -0.0125, 0.1875, 0.0875}}

```

Рис. 56. Время колебаний математического маятника

6.5 Изменение структуры списков

В табл. 18 приведем часто используемые функции, позволяющие изменять структуру списков.

Таблица 18. Функции для изменения структуры списков

$Sort[list, condition]$	Сортирует элементы списка $list$ согласно условию $condition$
$Sort[list, n, condition]$	Возвращает номер, на котором был бы расположен элемент на исходной позиции n в списке $list$ после его сортировки согласно условию $condition$
$Join[list_1, list_2, \dots]$	Объединяет списки $list_i$ в единую цепочку
$Union[list_1, list_2, \dots]$	Удаляет все повторения элементов, а сам список сортирует по возрастанию (объединение двух множеств)
$Intersection$ $[list_1, list_2, \dots]$	Находит пересечение нескольких множеств
$Complement$ $[list_1, list_2, \dots]$	Возвращает отсортированный список, состоящий из элементов, которые есть только в первом списке и не встречаются в остальных
$Reverse[list]$	Возвращает список, в котором элементы расположены в обратном порядке
$RotateLeft[list]$ $(RotateRight[list])$	Возвращает список, в котором элементы циклически сдвинуты на n позиций влево (вправо)

Для операций над структурами, содержащими в себе списки, существуют также удобные команды $Map[f, list]$ или $f/@list$ — применяет f к каждому элементу списка $list$; $Thread[f[arg]]$ — распараллеливает функцию f по любым спискам в аргументе arg .

Пример 31. Запишем уравнения Шредингера для трех стационарных состояний квантовой частицы с волновыми функциями

$\psi_1(x), \psi_2(x), \psi_3(x)$, как показано на рис. 57.

```

In[1]:= states = {ψ1[x], ψ2[x], ψ3[x]}
Out[1]= {ψ1[x], ψ2[x], ψ3[x]}

In[2]:= H[ψ_] := -ħ^2/(2*m)*D[ψ, {x, 2}] + V[x]*ψ
Out[2]= -ħ^2/(2*m)*D[ψ, {x, 2}] + V[x]*ψ

In[3]:= LeftPart = Map[H, states]
Out[3]= {V[x] ψ1[x] - (ħ^2 ψ1''[x])/2 m, V[x] ψ2[x] - (ħ^2 ψ2''[x])/2 m, V[x] ψ3[x] - (ħ^2 ψ3''[x])/2 m}

In[4]:= RightPart = ε*states
Out[4]= {ε ψ1[x], ε ψ2[x], ε ψ3[x]}

In[5]:= energyEquations = LeftPart == RightPart
Out[5]= {V[x] ψ1[x] - (ħ^2 ψ1''[x])/2 m == ε ψ1[x], V[x] ψ2[x] - (ħ^2 ψ2''[x])/2 m == ε ψ2[x], V[x] ψ3[x] - (ħ^2 ψ3''[x])/2 m == ε ψ3[x]}

In[6]:= Thread[energyEquations]
Out[6]= {V[x] ψ1[x] - (ħ^2 ψ1''[x])/2 m == ε ψ1[x], V[x] ψ2[x] - (ħ^2 ψ2''[x])/2 m == ε ψ2[x], V[x] ψ3[x] - (ħ^2 ψ3''[x])/2 m == ε ψ3[x]}

```

Рис. 57. Уравнения Шредингера для трех стационарных состояний

Пример 32. Рассмотрим 5 ближайших к Земле звезд (*NameNearest*) и 5 наиболее ярких (*NameBrightest*). Определим, какие звезды являются одновременно ближайшими и наиболее яркими (*NearAndBright*), какие из ближайших не принадлежат к самым ярким (*NearNotBright*). Выведем на экран спектральные классы звезд, к которым они относятся (*UniqueSpectralClasses*), а также рассчитаем среднюю температуру для всех 10 звезд, как продемонстрировано на рис. 58.

```

In[1]:= ListNearest =
  EntityClass["Star",
    {EntityProperty["Star", "DistanceFromEarth"] → TakeSmallest[5]}];
ListBrightest =
  EntityClass["Star",
    {EntityProperty["Star", "ApparentMagnitude"] → TakeSmallest[5]}];

In[3]:= NameNearest = EntityValue[ListNearest, "Name"]
Out[3]= {Sun, Proxima Centauri, Rigil Kentaurus, Toliman, Barnard's Star}

In[4]:= NameBrightest = EntityValue[ListBrightest, "Name"]
Out[4]= {Sun, Sirius, Canopus, Arcturus, Rigil Kentaurus}

In[5]:= NearAndBright = Intersection[NameNearest, NameBrightest]
Out[5]= {Rigil Kentaurus, Sun}

In[6]:= NearNotBright = Complement[NameNearest, NameBrightest]
Out[6]= {Barnard's Star, Proxima Centauri, Toliman}

In[7]:= SpectrNearest = EntityValue[ListNearest, "SpectralClass"]
Out[7]= {G2V, M5.5Ve, G2V, K1V, sdM4e}

In[8]:= SpectrBrightest = EntityValue[ListBrightest, "SpectralClass"]
Out[8]= {G2V, A1Vm, F0Ib, K2IIp, G2V}

In[9]:= UniqueSpectralClasses = Union[SpectrNearest, SpectrBrightest]
Out[9]= {A1Vm, F0Ib, G2V, K1V, K2IIp, M5.5Ve, sdM4e}

In[10]:= TempNearest =
  Sort[EntityValue[ListNearest, "EffectiveTemperature"], Greater]
Out[10]=
  { $5.8 \times 10^3$  K, 5772. K,  $5.0 \times 10^3$  K,  $3.04 \times 10^3$  K,  $2.8 \times 10^3$  K}

In[11]:= TempBrightest =
  Sort[EntityValue[ListBrightest, "EffectiveTemperature"], Greater]
Out[11]=
  {9940. K,  $7.5 \times 10^3$  K,  $5.8 \times 10^3$  K, 5772. K,  $4.4 \times 10^3$  K}

In[12]:= Mean[Join[TempNearest, TempBrightest]]
Out[12]=
   $5.6 \times 10^3$  K

```

Рис. 58. Свойства звезд

6.6 Векторы

Математическим векторам $(x_1, \dots, x_n)$ в Wolfram Language соответствуют списки $\{x_1, \dots, x_n\}$, где координаты вектора могут быть любыми выражениями. При действиях над векторами все обычные арифметические операции трактуются как покомпонентные, и при их выполнении скаляр x отождествляется с вектором $(x, \dots, x)$ подходящей длины. Основные функции для работы с векторами приведены в табл. 19.

Таблица 19. Функции для работы с векторами

<i>Norm</i> [<i>v</i>]	Расчет нормы вектора
<i>Projection</i> [<i>v</i> , <i>axis</i>]	Расчет проекции вектора <i>v</i> на выбранную ось <i>axis</i>
<i>Grad</i> [<i>v</i> , { <i>x</i> ₁ , <i>x</i> ₂ , ..., <i>x</i> _{<i>n</i>} }] (<i>Div</i> [<i>v</i> , { <i>x</i> ₁ , <i>x</i> ₂ , ..., <i>x</i> _{<i>n</i>} }], <i>Curl</i> [<i>v</i> , { <i>x</i> ₁ , <i>x</i> ₂ , ..., <i>x</i> _{<i>n</i>} }]	Расчет градиента (дивергенции, ротора) векторного поля, задаваемого вектором <i>v</i> в координатах { <i>x</i> ₁ , <i>x</i> ₂ , ..., <i>x</i> _{<i>n</i>} }
<i>VectorAngle</i> [<i>a</i> , <i>b</i>]	Расчет угла между векторами <i>a</i> и <i>b</i>
<i>Dot</i> [<i>a</i> , <i>b</i>] или <i>a.b</i>	Расчет скалярного произведения векторов (<i>a.b</i>), а также перемножение матриц или тензоров
<i>Cross</i> [<i>a</i> , <i>b</i>]	Расчет векторного произведения векторов [<i>a</i> × <i>b</i>]

Пример 33. Рассчитаем напряженность электрического поля, создаваемую в вершине равностороннего треугольника со стороной 1 м двумя точечными зарядами величиной -0.6 нКл и +0.4 нКл, расположенными в двух других вершинах, с помощью кода на рис. 59.

```

In[1]:= r = {x, y}
Out[1]= {x, y}

In[2]:= rn = Norm[r] /.
Assumptions -> {x ∈ Reals, y ∈ Reals} //
ComplexExpand
Out[2]=  $\sqrt{x^2 + y^2}$ 

In[3]:= efield = -Grad[k*q/rn, {x, y}]
Out[3]=  $\left\{ \frac{k q x}{(x^2 + y^2)^{3/2}}, \frac{k q y}{(x^2 + y^2)^{3/2}} \right\}$ 

In[4]:= triangle = ASATriangle[Pi/3, a, Pi/3]
Out[4]= Triangle[{{0, 0}, {a, 0},  $\left\{ \frac{a}{2}, \frac{\sqrt{3} a}{2} \right\}}$ }]

In[5]:= r1 = triangle[[1]][2] - triangle[[1]][1] /. a -> 1.
r2 = triangle[[1]][3] - triangle[[1]][1] /. a -> 1.
Out[5]= {1., 0}
Out[6]= {0.5, 0.866025}

In[7]:= efield1 =
efield /. {x -> r1[[1]], y -> r1[[2]],
k -> 8.9876*10^9, q -> -0.6*10^(-9)}
efield2 =
efield /. {x -> r2[[1]], y -> r2[[2]],
k -> 8.9876*10^9, q -> 0.4*10^(-9)}
Out[7]= {-5.39256, 0.}
Out[8]= {1.79752, 3.1134}

In[9]:= VectorAngle[efield1, efield2]*180/Pi
Out[9]= 120.

In[10]:= efield0 = efield1 + efield2
Out[10]= {-3.59504, 3.1134}

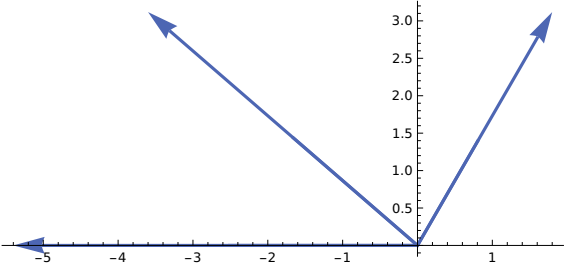
```

Рис. 59. Расчет напряженности электрического поля

Вначале найдем напряженности, создаваемые каждым из двух зарядов в заданной точке, $\vec{E}_1$ и $\vec{E}_2$, используя формулу $\vec{E} = -\nabla\varphi$ и зная выражение для потенциала поля точечного заряда на расстоянии r от него $\varphi = \frac{kq}{r}$. Рассчитаем угол между векторами $\vec{E}_1$ и $\vec{E}_2$. Применяя принцип суперпозиции, рассчитаем результирующую напряженность $\vec{E}_0 = \vec{E}_1 + \vec{E}_2$. Выведем на экран треугольник и вектора $\vec{E}_0$, $\vec{E}_1$ и $\vec{E}_2$ как на рис. 60.

```
In[11]:= qpoint = triangle[1][1] /. a -> 1.
Out[11]=
{0, 0}

In[12]:= plotef = ResourceFunction["PlotVector"][{efield1, efield2,
    efield0}, {qpoint, qpoint, qpoint}, VectorStyle -> Thick]
Out[12]=
```



```
In[13]:= Show[plotef, Graphics[{EdgeForm[{Thick, Dashed, Black}], Transparent,
    triangle /. a -> 1}]]
Out[13]=
```

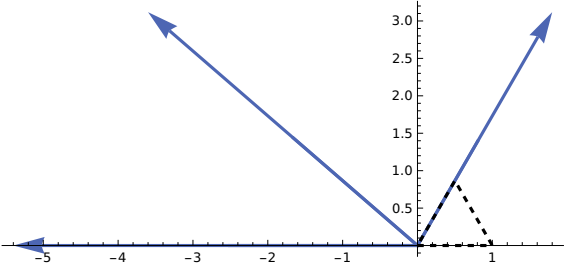


Рис. 60. Визуализация векторов напряженности $\vec{E}_0$, $\vec{E}_1$ и $\vec{E}_2$

Пример 34. Рассмотрим поле скоростей твердого тела $\vec{v} = \vec{\omega} \times \vec{r}$, вращающегося вокруг неподвижной точки с мгновенной угловой скоростью $\omega = (\omega_x, \omega_y, \omega_z)$, см. рис. 61. Отметим, что $\text{rot } \vec{v} = 2\vec{\omega}$.

```
In[1]:= w = {wx, wy, wz}
Out[1]= {wx, wy, wz}

In[2]:= v = Cross[w, {x, y, z}]
Out[2]= {-wz y + wy z, wz x - wx z, -wy x + wx y}

In[3]:= Curl[v, {x, y, z}]
Out[3]= {2 wx, 2 wy, 2 wz}

In[4]:= v1 = v /. {wx -> 1., wy -> 2., wz -> 1.}
Out[4]= {-1. y + 2. z, 1. x - 1. z, -2. x + 1. y}

In[5]:= VectorPlot3D[v1, {x, -3, 3}, {y, -3, 3}, {z, -3, 3}]
```

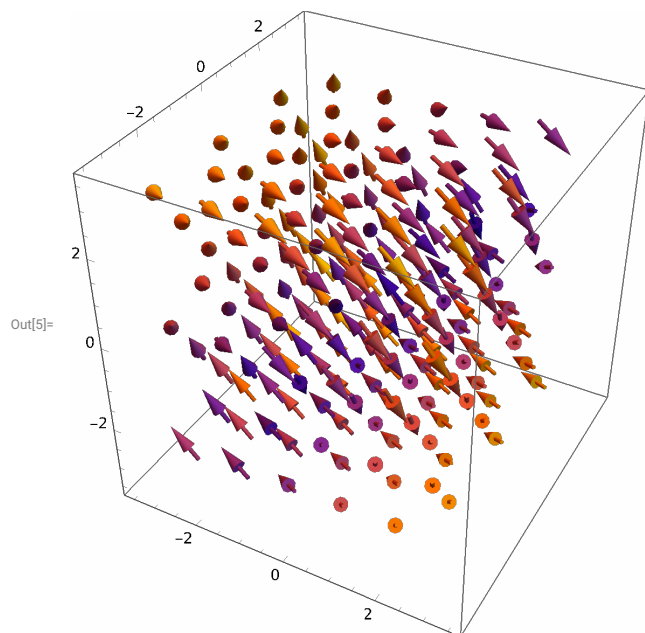


Рис. 61. Поле скоростей твердого тела

6.7 Матрицы

Списки списков одинаковой длины в Wolfram Language соответствуют матрицам. Матрица записывается как список из ее строк. Таким образом, матрица $\begin{pmatrix} a & b \\ c & d \end{pmatrix}$ вводится в виде $\{\{a, b\}, \{c, d\}\}$ или $List[List[a, b], List[c, d]]$. Для вывода матрицы в традиционной записи следует применить функцию *TableForm* или *MatrixForm*.

Пример 35. Запишем спиновые матрицы Паули $\sigma_1, \sigma_2, \sigma_3$ и проверим соотношение $\sigma_1 \cdot \sigma_2 = I\sigma_3$, как показано на рис. 62.

```
In[1]:= sigma1 = {{0, 1}, {1, 0}};
        sigma2 = {{0, -I}, {I, 0}};
        sigma3 = {{1, 0}, {0, -1}};
        MatrixForm[sigma1]
        MatrixForm[sigma2]
        MatrixForm[sigma3]

Out[4]//MatrixForm=

$$\begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$$


Out[5]//MatrixForm=

$$\begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}$$


Out[6]//MatrixForm=

$$\begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$$


In[7]:= sigma1.sigma2/I // MatrixForm

Out[7]//MatrixForm=

$$\begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$$

```

Рис. 62. Проверка соотношения $\sigma_1 \cdot \sigma_2 = I\sigma_3$ для матриц Паули

В системе Wolfram Mathematica существует набор команд генерации списков и матриц специального вида. *IdentityMatrix[n]* генерирует единичную матрицу $n \times n$, а *DiagonalMatrix[{d₁, ..., d_n}]* порождает диагональную матрицу с элементами $d_1, \dots, d_n$.

Пример 36. Сгенерируем метрический тензор в четырехмерном пространстве-времени Минковского и рассчитаем квадрат 4-импульса $p = (E, c\vec{p})$ частицы как показано на рис. 63.

```
In[1]:= P = {Et/c, px, py, pz};
MetricTensorM = DiagonalMatrix[{1, -1, -1, -1}];
fourMomentumLength = Sqrt[P . MetricTensorM . Transpose[P]]

Out[3]:=  $\sqrt{\frac{Et^2}{c^2} - px^2 - py^2 - pz^2}$ 
```

Рис. 63. Расчет квадрата 4-импульса $p = (E, c\vec{p})$ частицы

Части матриц извлекаются с помощью команд из табл. 20:

Таблица 20. Извлечение частей матриц

$x[[i, j]]$	Элемент x в позиции (i, j)
$x[[\{i, j\}, \{k, l\}]]$	Подматрица порядка 2 матрицы x , стоящая на пересечении строк с номерами i, j и столбцов с номерами k, l
$x[[i]]$	i -я строка матрицы x
$x[[All, j]]$	j -й столбец матрицы x
$Tr[x]$	След матрицы x
$Tr[x, List]$	Главная диагональ матрицы x
$Det[x]$	Определитель квадратной матрицы x

В примере 36 использована команда *Transpose*, переводящая матрицу в транспонированную матрицу, строки которой совпадают со столбцами исходной. По умолчанию команда *Transpose* представляет два верхних уровня вложенности списков. Поэтому для одновременного транспонирования списка матриц нужно применять *Transpose* к элементам этого списка при помощи команды *Map*.

Для заданной матрицы можно найти матрицу, обратную ей, а также собственные вектора (СВ) и собственные значения (СЗ) с помощью функций из табл. 21.

Таблица 21. Действия с матрицами

$Inverse[A]$	Матрица, обратная матрице A
$Eigenvalues[A]$	СЗ матрицы A
$Eigenvectors[A]$	СВ матрицы A
$Eigensystem[A]$	СЗ и СВ матрицы A , ответ представляется в виде набора СЗ и соответствующих им СВ $\{\{\lambda_1, \lambda_2, \dots\}, \{\{x_1, x_2, \dots\}, \{y_1, y_2, \dots\}, \dots\}\}$

Пример 37. Рассчитаем СЗ матрицы упругих постоянных (тензора упругости) кубического кристалла C_{ij} , $i, j = 1, \dots, 6$ и соответствующий тензор податливости $S_{ij} = (C_{ij}^{-1})$, см. рис. 64.

```
In[1]:= Elastic = {
    {c11, c12, c12, 0, 0, 0},
    {c12, c11, c12, 0, 0, 0},
    {c12, c12, c11, 0, 0, 0},
    {0, 0, 0, c44, 0, 0},
    {0, 0, 0, 0, c44, 0},
    {0, 0, 0, 0, 0, c44}
};

In[2]:= eigenvals = Eigenvalues[Elastic]
Out[2]:= {c11 - c12, c11 - c12, c11 + 2 c12, c44, c44, c44}

In[3]:= S = Inverse[Elastic] /. {c11 -> 49., c12 -> 12., c44 -> 12.} // MatrixForm
Out[3]//MatrixForm=
```

$$\begin{pmatrix} 0.0225842 & -0.0044428 & -0.0044428 & 0 & 0 & 0 \\ -0.0044428 & 0.0225842 & -0.0044428 & 0 & 0 & 0 \\ -0.0044428 & -0.0044428 & 0.0225842 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.0833333 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.0833333 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0.0833333 \end{pmatrix}$$

Рис. 64. Расчет СЗ тензора упругости и обратного ему тензора

При действиях над матрицами все обычные операции трактуются как покомпонентные. Также матрицы можно создавать из векторов или более глубоких списков при помощи структурных команд, таких, как *Partition* или *Flatten*.

Пример 38. Рассчитаем первые три инварианта I_1, I_2, I_3 тензора механических напряжений $\sigma_{ij}, i, j = x, y, z$: $I_1 = \text{tr}(\sigma) = \sigma_{xx} + \sigma_{yy} + \sigma_{zz}$, $I_2 = \frac{1}{2} (\text{tr}^2(\sigma) - \text{tr}(\sigma^2))$, $I_3 = \det(\sigma)$, см. рис. 65.

```

In[1]:= sigma = {{σxx, σxy, σxz},
                 {σyx, σyy, σyz},
                 {σzx, σzy, σzz}};
I1 = Tr[sigma]

Out[2]= σxx + σyy + σzz

In[3]:= subsigma1 = {{sigma[[2, 2]], sigma[[2, 3]], {sigma[[3, 2]], sigma[[3, 3]]}};
subsigma2 = {{sigma[[1, 1]], sigma[[1, 2]], {sigma[[2, 1]], sigma[[2, 2]]}};
subsigma3 = {{sigma[[1, 1]], sigma[[1, 3]], {sigma[[3, 1]], sigma[[3, 3]]}};
det1 = Det[subsigma1];
det2 = Det[subsigma2];
det3 = Det[subsigma3];

In[9]:= I2 = (1/2) (Tr[sigma]^2 - Tr[sigma.sigma])

Out[9]= 1/2 (-σxx^2 - 2 σxy σyx - σyy^2 - 2 σxz σzx - 2 σyz σzy - σzz^2 + (σxx + σyy + σzz)^2)

In[10]:= I20 = det1 + det2 + det3

Out[10]= -σxy σyx + σxx σyy - σxz σzx - σyz σzy + σxx σzz + σyy σzz

In[11]:= I2 - I20 // Simplify

Out[11]= 0

In[12]:= I3 = Det[sigma]

Out[12]= -σxz σyy σzx + σxy σyz σzx + σxz σyx σzy - σxx σyz σzy - σxy σyx σzz + σxx σyy σzz

In[13]:= {I1, I2, I3} /. {σxx → 50., σxy → 30., σxz → 20.,
                        σyx → 30., σyy → -20., σyz → -10.,
                        σzx → 20., σzy → -10., σzz → 10.}

Out[13]= {40., -2100., -28 000.}

```

Рис. 65. Расчет трех первых инвариантов тензора напряжений

6.8 Массивы

В языке Wolfram Language массивы — это списки, в которых каждый элемент сам по себе является списком. Возможны и гораздо более общие структуры, сочетающие другие элементы. Для задания массивов используются функции, приведенные в табл. 22.

Таблица 22. Функции для задания массивов

$Array[f, n]$	Генерирует список длиной n с элементами $f[i]$, где i имеет значения от 1 до n
$Array[f, \{n_1, n_2, \dots\}]$	Генерирует массив с размерностью $n_1 \times n_2, \dots$ в виде вложенных списков с элементами $f[i_1, i_2, \dots]$
$SparseArray[\{pos_1 \rightarrow f_1, pos_2 \rightarrow f_2, \dots\}]$	Генерирует разреженный массив, где все элементы нулевые, за исключением значений f_i в позициях pos_i

Пример 39. Рассмотрим массы, периоды обращения и эксцентриситеты орбит планет Солнечной системы. Представим полученную информацию графически с помощью функции *ArrayPlot*, как показано на рис. 66.

```
In[1]:= PlanetArray = Log[QuantityMagnitude[{
UnitConvert[PlanetData[All, "Mass"], "EarthMass"],
UnitConvert[PlanetData[All, "OrbitPeriod"], "Years"],
PlanetData[All, "Eccentricity"]}]];
PlanetArray // MatrixForm
```

Out[2]//MatrixForm=

-2.895	-0.205	$0. \times 10^{-4}$	-2.231	5.762	4.556	2.677	2.842
-1.422910	-0.4851276	0.0007021	0.6324072	2.4740766	3.3832936	4.4317020	5.1053646
-1.5816735	-4.99478	-4.0917348	-2.3707319	-3.0284071	-2.91599	-3.0540457	-4.75764

```
In[3]:= ArrayPlot[PlanetArray, Mesh → True, ColorFunction → "GrayTones"]
```

Out[3]=

Рис. 66. Использование *ArrayPlot* для визуализации массивов

Пример 40. Выведем на экран таблицу, содержащую электрические заряды и токовые массы кварков, как показано на рис. 67.

```
In[1]:= quarkNames = {"UpQuark", "DownQuark", "CharmQuark",
    "StrangeQuark", "TopQuark", "BottomQuark"}
Out[1]= {UpQuark, DownQuark, CharmQuark, StrangeQuark, TopQuark, BottomQuark}

In[2]:= qData = Array[Function[i, { quarkNames[[i]],
    ParticleData[quarkNames[[i]], "Charge"],
    ParticleData[quarkNames[[i]], "Mass"]}],
    Length[quarkNames]];

In[3]:= qData // TableForm
Out[3]//TableForm=
```

UpQuark	$\frac{2}{3}$	2.2 MeV/c ²
DownQuark	$-\frac{1}{3}$	5.0 MeV/c ²
CharmQuark	$\frac{2}{3}$	1250. MeV/c ²
StrangeQuark	$-\frac{1}{3}$	95. MeV/c ²
TopQuark	$\frac{2}{3}$	174 200. MeV/c ²
BottomQuark	$-\frac{1}{3}$	4200. MeV/c ²

Рис. 67. Использование массивов для создания таблицы данных

7 Последовательности, суммы и произведения

7.1 Последовательности, ряды и пределы

В языке Wolfram Language целочисленные последовательности представлены списками. Для генерации последовательности по известной формуле $f(n)$ используется функция *Table*. Обратное действие, т.е. нахождение генерирующей функции для известной последовательности, выполняется с помощью команды *FindSequenceFunction* согласно рис. 68.

```
In[1]:= Table[n^2, {n, 1, 8}]
Out[1]= {1, 4, 9, 16, 25, 36, 49, 64}

In[2]:= FindSequenceFunction[Out[1], n]
Out[2]= n^2
```

Рис. 68. Генерация последовательности

Некоторые хорошо известные последовательности определяются встроенными функциями, например, последовательность чисел Фибоначчи *Fibonacci[n]*. Она является рекуррентной, в которой каждый следующий член выражается через предыдущие по формулам $f_1 = f_2 = 1, f_n = f_{n-1} + f_{n-2}$. Для задания рекуррентных последовательностей существует функция *RecurrenceTable*.

Пример 41. Выведем первые 10 членов последовательности Фибоначчи двумя способами и вычислим их сумму, см. рис. 69. Рассчитаем предел отношения двух соседних чисел последовательности Фибоначчи с помощью функции *DiscreteLimit* и сравним полученное значение предела с известным значением золотого сечения на рис. 70. Графическое представление в виде известной спирали Фибоначчи можно получить с помощью кода на рис. 71.

```

In[1]:= Table[Fibonacci[x], {x, 1, 10}]
Out[1]:= {1, 1, 2, 3, 5, 8, 13, 21, 34, 55}

In[2]:= RecurrenceTable[{f[n] == f[n - 1] + f[n - 2],
    f[1] == 1, f[2] == 1}, f, {n, 10}]
Out[2]:= {1, 1, 2, 3, 5, 8, 13, 21, 34, 55}

In[3]:= Total[%]
Out[3]:= 143

```

Рис. 69. Последовательность Фибоначчи

```

In[1]:= Limf = DiscreteLimit[Fibonacci[n + 1]/Fibonacci[n],
    n -> Infinity]
Out[1]:=  $\frac{1}{2} (1 + \sqrt{5})$ 

In[2]:= N[limf] === N[GoldenRatio]
Out[2]:= True

```

Рис. 70. Расчет золотого сечения

```

In[1]:= With[{tf = AffineTransform[{
    RotationMatrix[-Pi/2]/GoldenRatio, {1, 1}]}],
    Graphics[{{NestList[tf, RegionBoundary@Rectangle[], 9]},
    Thick, NestList[tf,
    Circle[{(1, -1) + 1}/2, 1, {-Pi/4, Pi/4} + 3 Pi/4], 9]},
    PlotRange -> {{-0.1, 1.7}, {-0.1, 1.1}}]]

```

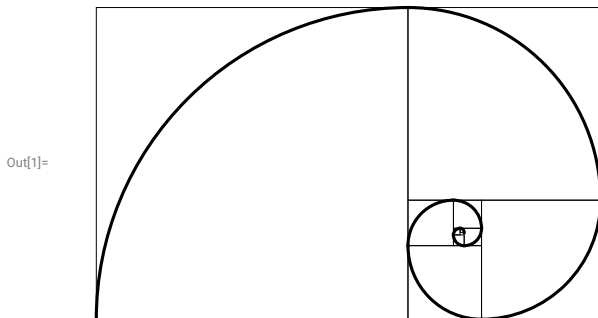


Рис. 71. Спираль Фибоначчи

Пример 42. Аналогичным образом рассчитаем первый замечательный предел $\lim_{n \rightarrow \infty} \left(1 + \frac{1}{n}\right)^n = e$, см. рис. 72.

```
In[1]:= lim1 = DiscreteLimit[(1 + 1/n)^n, n -> Infinity]
Out[1]= e
```

Рис. 72. Расчет первого замечательного предела

Числовая последовательность, члены которой соединены знаком $+$, называется рядом. Для вычисления конечных и бесконечных сумм рядов используется функция *Sum*. Она позволяет вычислять суммы символьных и числовых рядов, причем один или оба предела суммирования могут быть бесконечными либо символами. Способы указания диапазонов переменных в *Sum*, представленные в табл. 23, являются примером общей нотации итератора, которую использует язык Wolfram Language, и встречаются, в частности, в функциях *Product*, *Table*, а также циклах *Do*. Стандартный символ суммы набирается комбинацией клавиш *Esc* + *sumt* + *Esc*.

Таблица 23. Способы использования функции *Sum*

$Sum[f[i], \{i, i_{max}\}]$	Вычисляет сумму значений f при изменении индекса i от 1 до i_{max} с шагом 1
$Sum[f[i], \{i, i_{min}, i_{max}\}]$	Вычисляет сумму значений f при изменении индекса i от i_{min} до $i = i_{max}$ с шагом 1
$Sum[f[i], \{i, i_{min}, i_{max}, di\}]$	Действует аналогично предыдущему с шагом di
$Sum[f[i], \{i, i_{min}, i_{max}\}, \{j, j_{min}, j_{max}\}, \dots]$	Вычисляет многократную сумму значений f по переменным $i, j, \dots$, где число индексных переменных неограничено.

Пример 43. На рис. 73 рассчитаем сумму геометрической прогрессии $\sum_{n=1}^{\infty} q^n$ с выводом условий справедливости решения.

```
In[1]:= Sum[a*q^n, {n, 0, Infinity}, GenerateConditions -> True]
```

```
Out[1]:=  $\frac{a}{1-q}$  if Abs[q] < 1
```

Рис. 73. Расчет суммы геометрической прогрессии

Пример 44. Рассчитаем сумму арифметической прогрессии в виде ряда нечетных чисел согласно рисунку 74.

```
In[1]:= Sum[i, {i, 1, 2*n-1, 2}]
Out[1]:= Floor[n]^2
```

```
In[2]:= Sum[2^i / Factorial[i], {i, 1, n}]
Out[2]:=  $-1 + \frac{2^n}{n!} + \frac{e^2 \Gamma[n, 2]}{\Gamma[n]}$ 
```

Рис. 74. Расчет суммы арифметической прогрессии

Здесь $\Gamma(x, y) = \int_y^{\infty} t^{x-1} e^{-t} dt$ — неполная гамма-функция, а $\Gamma[x]$ — обычная гамма-функция Эйлера. Аналогичным образом многие бесконечные суммы выражаются как значения элементарных или специальных функций, см. рис. 75, где *HypergeometricPFQ* — обобщенная гипергеометрическая функция, $\zeta[3]$ — дзета-функция Римана.

```
In[1]:= Sum[1/(n!*x^n), {n, 0, Infinity}]
Out[1]:=  $e^{\frac{1}{x}}$ 
```

```
In[2]:= Sum[1/n^3, {n, 1, Infinity}]
Out[2]:= Zeta[3]
```

```
In[3]:= Sum[1/(n!*n^3), {n, 1, Infinity}]
Out[3]:= HypergeometricPFQ[{1, 1, 1, 1}, {2, 2, 2, 2}, 1]
```

Рис. 75. Представления некоторых бесконечных сумм

Для численного расчета сумм используется функция *NSum* с аналогичным синтаксисом, которая является предпочтительной, если изначально значение суммы нужно получить не в аналитическом, а в численном виде. Данная функция оптимизирована для численных расчетов, что ускоряет получение результата по сравнению с *Sum*.

Пример 45. Рассчитаем сумму энергий 11 первых уровней квантового гармонического осциллятора, для энергии n -го уровня которого справедлива формула: $E_n = (n + \frac{1}{2}) \hbar \omega$, где $n = 0, 1, 2, \dots$, $\hbar$ — постоянная Планка, ω — угловая частота осциллятора. Выберем $\omega = 1$ Гц, тогда код может быть записан согласно рис. 76:

```
In[1]:= hbar = 1.0545718*10^-34;
        omega = 1.0;
        nMax = 10;
        energySum = NSum[hbar * omega * (n + 1/2), {n, 0, nMax}]
Out[4]= 6.38016*10^-33
```

Рис. 76. Уровни энергии квантового гармонического осциллятора

7.2 Произведения

Функции *Product* и *NProduct* позволяют вычислять произведения в аналитическом и численном виде, соответственно. Правила записи аргументов и возможности вычислений те же, что и для *Sum* и *NSum*, а именно, *Product*[$f[i]$, { i, m, n }] выражает произведение $\prod_{i=m}^n f(i)$ значений функции f по i от m до n с шагом 1. Как и в случае сумм, один или оба предела могут быть бесконечными.

Пример 46. Вычислим эйлерово представление суммы ряда обратных квадратов, см. рис. 77, где *Prime*[n] — n -е простое число.

```
In[1]:= Product[(1 - 1/Prime[n]^2)^(-1), {n, 1, Infinity}]
Out[1]=  $\frac{\pi^2}{6}$ 
```

Рис. 77. Вычисление суммы ряда обратных квадратов

8 Ряды и пределы

Степенные ряды во многом являются алгебраическим аналогом чисел ограниченной точности. Язык Wolfram Language поддерживает не только обычные степенные ряды, но также ряды Лорана и ряды Пуансо, а также сложные асимптотические разложения для специальных функций со сложными структурами разрезом. Пакет Wolfram Mathematica позволяет выполнить разложение в ряд до членов n -го порядка по степеням малого параметра $(x - x_0)$ любых функций, заданных аналитически, командой $Series[F, \{x, x_0, n\}]$.

Пример 47. Приведем разложение в ряд экспоненциальной функции e^{ikx} и потенциала Юкавы $V(r) = -g^2 \frac{e^{-mr}}{r}$, где g — константа взаимодействия, m — масса частицы-переносчика взаимодействия, как продемонстрировано на рис. 78.

```
In[1]:= expFunction = Exp[I k x];

In[2]:= seriesExpansionExp = Series[expFunction, {k, 0, 5}]

Out[2]= 1 + i x k -  $\frac{x^2 k^2}{2}$  -  $\frac{1}{6} i x^3 k^3$  +  $\frac{x^4 k^4}{24}$  +  $\frac{1}{120} i x^5 k^5$  + O[k]^6

In[3]:= Normal[seriesExpansionExp]

Out[3]= 1 + i k x -  $\frac{k^2 x^2}{2}$  -  $\frac{1}{6} i k^3 x^3$  +  $\frac{k^4 x^4}{24}$  +  $\frac{1}{120} i k^5 x^5$ 

In[4]:= YukawaPotential[g_, m_, r_] := -g^2 (Exp[-m r]/r);

In[5]:= seriesExpansionYukawa = Series[YukawaPotential[g, m, r], {r, 0, 2}]

Out[5]= - $\frac{g^2}{r}$  +  $g^2 m$  -  $\frac{1}{2} (g^2 m^2) r$  +  $\frac{1}{6} g^2 m^3 r^2$  + O[r]^3
```

Рис. 78. Разложение в ряды

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

Для случая неизвестной или неопределенной функции *Series* возвращает степенной ряд в виде производных, см. рис. 79.

```
In[1]:= Series[2 f[x] - 3, {x, a, 3}]
Out[1]= (-3 + 2 f[a]) + 2 f'[a] (x - a) + f''[a] (x - a)^2 + 1/3 f^(3)[a] (x - a)^3 + O[x - a]^4
```

Рис. 79. Степенной ряд в виде производных

Основной командой для вычисления пределов аналитически заданных функций является *Limit*. Для вычисления $\lim_{x \rightarrow c} f(x)$ следует набрать *Limit*[*f*[*x*], *x* $\rightarrow$ *c*], как продемонстрировано на рис. 80.

```
In[2]:= Limit[(Cos[m * x] - Cos[n * x]) / (x ^ 2), x -> 0]
Out[2]= 1/2 (-m^2 + n^2)
```

Рис. 80. Пример работы функции *Limit*

Для вычисления левого $\lim_{x \rightarrow c-} f(x)$ и правого предела $\lim_{x \rightarrow c+} f(x)$ команда *Limit* вызывается в форматах: *Limit*[*f*[*x*], *x* $\rightarrow$ *c*, *Direction* $\rightarrow$ 1] и *Limit*[*f*[*x*], *x* $\rightarrow$ *c*, *Direction* $\rightarrow$ -1], что равнозначно *Limit*[*f*[*x*], *x* $\rightarrow$ *c*, *Direction* $\rightarrow$ "FromAbove"] и *Limit*[*f*[*x*], *x* $\rightarrow$ *c*, *Direction* $\rightarrow$ "FromBelow"], где опция *Direction* определяет, слева или справа берется предел, см. рис. 81.

```
In[1]:= Limit[Sin[x]/x, x -> 0]
Out[1]= 1

In[2]:= Limit[Tan[x], x -> Pi/2, Direction -> 1]
Out[2]= ∞

In[3]:= Limit[Tan[x], x -> Pi/2, Direction -> -1]
Out[3]= -∞

In[4]:= Limit[RealAbs[x]/x, x -> 0, Direction -> -1]
Out[4]= 1
```

Рис. 81. Вычисление левого и правого предела

9 Полиномы и рациональные функции

В системе Wolfram Mathematica есть набор удобных команд для работы с полиномами, приведенных в табл. 24.

Таблица 24. Функции для работы с полиномами

<i>Expand</i> [<i>poly</i>]	Раскладывает полином
<i>Factor</i> [<i>poly</i>]	Факторизует полином
<i>FactorTerms</i> [<i>poly</i>]	Выносит за скобки общий множитель.
<i>Collect</i> [<i>poly</i> , <i>x</i>]	Представляет данное выражение как полином по степеням <i>x</i>
<i>PolynomialGCD</i> [<i>poly1</i> , <i>poly2</i> , ...]	Находит общий делитель полиномов
<i>PowerExpand</i> [...]	Упрощает степенные выражения вида $\sqrt{x^2} \rightarrow x$.
<i>Coefficient</i> [<i>poly</i> , <i>form</i> , <i>n</i>]	Выводит коэффициент перед выражением вида <i>form</i> .

Для работы с рациональными дробями в системе Wolfram Mathematica существуют функции, приведенные в табл. 25.

Таблица 25. Функции для работы с рациональными дробями

<i>Together</i> [<i>expr</i>]	Приводит к общему знаменателю
<i>Apart</i> [<i>expr</i>]	Разлагает на простые дроби
<i>Cancel</i> [<i>expr</i>]	Сокращает общие множители в числителе и знаменателе <i>expr</i> .

Пример 48. Рассмотрим полиномы Лагерра, которые определяются формулой $L_n(x) = \frac{e^x}{n!} \frac{d^n}{dx^n}(x^n e^{-x})$, $x \geq 0$ или вычисляются по рекуррентной формуле для $n \geq 1$, где $L_1 = 1$: $L_{n+1}(x) = \frac{1}{n+1} ((2n+1-x)L_n(x) - nL_{n-1}(x))$. Проверим справедливость данной формулы для $n = 4$ с помощью кода на рис. 82.

Пример 49. Покажем для частного случая n , что полиномы Эрмита, определяемые формулой $H_n(x) = (-1)^n e^{x^2} \frac{d^n}{dx^n} e^{-x^2}$, явля-

ются нечетными функциями x ; проверим справедливость формулы $H_{n+1}(x) - 2xH_n(x) + 2nH_{n-1}(x) = 0$, см. рис. 83.

```
In[1]:= n = 4
Out[1]= 4

In[2]:= lhs = LaguerreL[n + 1, x]
Out[2]=  $\frac{1}{120} (120 - 600 x + 600 x^2 - 200 x^3 + 25 x^4 - x^5)$ 

In[3]:= rhs = (2 * n + 1 - x) * LaguerreL[n, x] - n * LaguerreL[n - 1, x]
Out[3]=  $-\frac{2}{3} (6 - 18 x + 9 x^2 - x^3) + \frac{1}{24} (9 - x) (24 - 96 x + 72 x^2 - 16 x^3 + x^4)$ 

In[4]:= rhs = rhs / (n + 1)
Out[4]=  $\frac{1}{5} \left( -\frac{2}{3} (6 - 18 x + 9 x^2 - x^3) + \frac{1}{24} (9 - x) (24 - 96 x + 72 x^2 - 16 x^3 + x^4) \right)$ 

In[5]:= lhsExpanded = Expand[lhs]
Out[5]=  $1 - 5 x + 5 x^2 - \frac{5 x^3}{3} + \frac{5 x^4}{24} - \frac{x^5}{120}$ 

In[6]:= rhsExpanded = Expand[rhs]
Out[6]=  $1 - 5 x + 5 x^2 - \frac{5 x^3}{3} + \frac{5 x^4}{24} - \frac{x^5}{120}$ 

In[7]:= lhsExpanded == rhsExpanded
Out[7]= True
```

Рис. 82. Проверка рекуррентной формулы для полиномов Лагерра

```
In[1]:= n = 10
Out[1]= 10

In[2]:= Apart[HermiteH[n, x]/HermiteH[n, -x]]
Out[2]= 1

In[3]:= Apart[HermiteH[n + 1, x]/HermiteH[n + 1, -x]]
Out[3]= -1

In[4]:= HermiteH[n + 1, x] - 2 * x * HermiteH[n, x] + 2 * n * HermiteH[n - 1, x] // Simplify
Out[4]= 0
```

Рис. 83. Исследование свойств полиномов Эрмита

10 Алгебраические уравнения

Уравнения в Wolfram Mathematica записываются в виде $f[x]==g[x]$. Для точного решения уравнений в системе реализованы функции из табл. 26.

Таблица 26. Функции для решения алгебраических уравнений

$Roots[f == g, x]$	Решает полиномиальное уравнение $f(x) = g(x)$
$Solve[f == g, x]$	Решает уравнение $f(x) = g(x)$ относительно x
$Reduce[f == g, x]$	Упрощает уравнение $f(x) = g(x)$

Для приближенного вычисления корней служат функции $NRoots$, $NSolve$ и $FindRoot$, см. табл. 27.

Таблица 27. Численное решение алгебраических уравнений

$NRoots[f == g, x]$	Численно решает полиномиальное уравнение $f(x) = g(x)$
$NSolve[f == g, x]$	Численно решает уравнение $f(x) = g(x)$
$FindRoot[f == g, x, x_0]$	Ищет приближенное численное решение для уравнения $f(x) = g(x)$ начиная процедуру приближения с точки x_0

Пример 50. Найдем дальность полета снаряда из орудия, расположенного на высоте h под углом α к горизонту, зная закон движения снаряда, $x(t) = v_0 t \cos \alpha$, $y(t) = h + v_0 t \sin \alpha - \frac{gt^2}{2}$, как на рис. 84.

```

In[1]:= y = h + v0*t*Sin[alpha] - (1/2)g*t^2;
timeOfFlight = Solve[y == 0, t, Reals,
Assumptions -> {g > 0, v0 > 0, h > 0, 0 < alpha < Pi/2, t > 0}]
Out[23]=

$$\left\{ \left\{ t \rightarrow \frac{v_0 \sin[\alpha]}{g} + \frac{\sqrt{2 g h + v_0^2 \sin[\alpha]^2}}{g} \right\} \right\}$$


In[26]:= tflight = t /. timeOfFlight[[1]]
Out[26]=

$$\frac{v_0 \sin[\alpha]}{g} + \frac{\sqrt{2 g h + v_0^2 \sin[\alpha]^2}}{g}$$


In[27]:= x = v0*tflight * Cos[alpha]
Out[27]=

$$v_0 \cos[\alpha] \left( \frac{v_0 \sin[\alpha]}{g} + \frac{\sqrt{2 g h + v_0^2 \sin[\alpha]^2}}{g} \right)$$


```

Рис. 84. Нахождение дальности полета снаряда

Команду *Solve* можно применять для решения системы уравнений относительно нескольких переменных. При этом как уравнения, так и неизвестные, относительно которых эти уравнения решаются, можно задавать в форме списка:

$$\text{Solve}[\{f_1 == g_1, \dots, f_m == g_m\}, \{x_1, \dots, x_n\}],$$

что эквивалентно использованию логических операторов $f_1 == g_1 \&\& f_2 == g_2$.

Пример 51. Рассмотрим абсолютно упругий центральный удар двух шаров массы m_1 и m_2 , исходно движущихся навстречу друг другу со скоростями v_{1i} и v_{2i} относительно Земли. Найдем их скорости после удара, v_{1f} и v_{2f} . Для этого запишем законы сохранения

импульса и энергии:

$$\begin{aligned} m_{1i}v_{1i} + m_{2i}v_{2i} &= m_{1f}v_{1f} + m_{2f}v_{2f} \\ \frac{m_{1i}v_{1i}^2}{2} + \frac{m_{2i}v_{2i}^2}{2} &= \frac{m_{1f}v_{1f}^2}{2} + \frac{m_{2f}v_{2f}^2}{2} \end{aligned}$$

и решим полученную систему уравнений с помощью *Solve*. Результат сравним с формульными данными системы Wolfram Mathematica, как показано на рис. 85.

```
In[1]:= Solve[{m1*v1i + m2*v2i == m1*v1f + m2*v2f,
  m1*v1i^2 + m2*v2i^2 == m1*v1f^2 + m2*v2f^2}, {v1f, v2f},
  Assumptions -> {v1f != v1i, v2f != v2i}]

Out[1]:= {{v1f -> (m1 v1i - m2 v1i + 2 m2 v2i) / (m1 + m2), v2f -> (2 m1 v1i - m1 v2i + m2 v2i) / (m1 + m2)}}

In[2]:= FormulaData["ElasticCollision"]

Out[2]:= {v1,f == (m1 v1,i + m2 v2,i + m2 (-v1,i + v2,i)) / (m1 + m2), v2,f == (m1 v1,i + m1 (v1,i - v2,i) + m2 v2,i) / (m1 + m2)}
```

Рис. 85. Расчет скоростей шаров после упругого удара

Команда *Solve* порождает только общие решения и игнорирует решения, возникающие при специальных значениях параметров, выдавая ответ вместе с сообщением о возможной ошибке. В таких случаях нужно применять команду *Reduce*. Она является самой мощной командой для решения уравнения $f(x) = g(x)$, так как команда *Reduce[expr, x]* пытается упростить любое выражение *expr* от *x*, содержащее равенства, неравенства, включения, логические связи и кванторы. В случае, когда эти выражения состоят только из уравнений и неравенств, команда *Reduce* пытается их решить. В отличие от команды *Solve*, она может указать все решения уравнения, даже если этих решений бесконечно много.

Пример 52. Рассчитаем значения углов, под которыми на экране

видны главные интерференционные максимумы при прохождении света с длиной волны λ через дифракционную решетку с периодом d . Для этого запишем формулу для главных максимумов дифракционной решетки $d \sin(\theta_m) = m\lambda$, где $m = 0, 1, 2, \dots$ и воспользуемся функциями *Solve* и *Reduce*, как показано на рис. 86.

```
In[1]:= Solve[Sin[θ] == m*λ/d, θ]

Out[1]:= {{θ → ArcSin[ $\frac{m\lambda}{d}$ ], {θ →  $\pi - \text{ArcSin}[\frac{m\lambda}{d}]$  if  $c_1 \in \mathbb{Z}$ }}, {θ →  $\text{ArcSin}[\frac{m\lambda}{d}] + 2\pi c_1$  if  $c_1 \in \mathbb{Z}$ }}
```

```
In[2]:= Reduce[Sin[θ] == m*λ/d, θ]

Out[2]:=  $c_1 \in \mathbb{Z} \ \&\& \ d \neq 0 \ \&\& \ \left( \theta == \pi - \text{ArcSin}[\frac{m\lambda}{d}] + 2\pi c_1 \parallel \theta == \text{ArcSin}[\frac{m\lambda}{d}] + 2\pi c_1 \right)$ 
```

Рис. 86. Расчет главных интерференционных максимумов

Пример 53. Вычислим нули волновой функции электрона в атоме водорода в состоянии с $n = 5, l = 1, m = 0$. Из курса квантовой физики известно, что волновая функция электрона в атоме водорода имеет вид:

$$\phi_{n,l,m}(r, \theta, \phi) = R_{n,l}(r)Y_l^m(\theta, \phi),$$

где $R_{n,l}$ и $Y_l^m(\theta, \phi)$ — решения радиальной и угловой частей уравнения Шредингера, n, l, m — главное, орбитальное и магнитное квантовые числа электрона, r, θ, ϕ — сферические координаты. Радиальная часть волновой функции записывается в виде

$$R_{n,l} = \sqrt{\frac{(n-1-l)!}{2n[(n+1)!]^3}} \left(\frac{2}{na_0}\right)^{l+3/2} r^l \exp\left(-\frac{r}{na_0}\right) L_{n+1}^{2l+1}\left(\frac{2r}{na_0}\right),$$

где L_{n+1}^{2l+1} — присоединенный полином Лагерра, a_0 — боровский радиус. Для всех $l \neq 0$ нули функции $R_{n,l}$ определяются очевидным решением $r = 0$ и корнями полиномиального уравнения $L_{n+1}^{2l+1}\left(\frac{2r}{na_0}\right) = 0$, которое можно решить с помощью функции

NRoots. В отличие от функции *NRoots*, функция *NSolve* может быть использована для численного нахождения корней произвольных трансцендентных уравнений, содержащих тригонометрические и показательные функции. Для иллюстрации корректности решения выведем на экран график функции $R_{n,l}$ для $n = 5, l = 1$, как показано на рис. 87.

```

In[1]:= F[n_, l_, x_] := x^l * Exp[-x/2] * LaguerreL[n - l - 1, 2 l + 1, x];
R[n_, l_, r_] := (2/(n*a))^(3/2) *
  Sqrt[Factorial[n - l - 1]/(2*n*Factorial[l])] *
  F[n, l, 2 r/(n a)];
a = 1.;

In[4]:= NRoots[LaguerreL[n - l - 1, 2 l + 1, 2 r/(n a)] == 0 /. {n -> 5, l -> 2}, r]
Out[4]= r == 10.8856 || r == 24.1144

In[5]:= NSolve[R[n, l, r] /. {n -> 5, l -> 2}, r]
NSolve: Inverse functions are being used by NSolve, so some solutions may not be found; use Reduce for
complete solution information.

Out[5]= {{r -> 0.}, {r -> 10.8856}, {r -> 24.1144}}

In[6]:= Plot[R[5, 2, r], {r, 0, 60}, PlotStyle -> Thick, LabelStyle -> Medium]
Out[6]=

```

Рис. 87. Радиальная волновая функция электрона

Обратим внимание, что результат работы команды *NSolve* был получен с предупреждением: *NSolve* использует обратные функции, поэтому некоторые решения могут быть не найдены; используйте *Reduce* для полной информации о решении.

Сферические гармоники $Y_l^m(\theta, \varphi)$ выражаются формулой:

$$Y_l^m(\theta, \varphi) = \sqrt{\frac{(2l+1)}{(4\pi)}} \sqrt{\frac{(l-m)!}{(l+m)!}} P_l^m(\cos(\theta)) \exp(im\varphi),$$

где P_l^m — присоединенные полиномы Лежандра. Для нахождения нулей функции $Y_5^1(\theta, \varphi)$ воспользуемся командами *NSolve* и *Reduce*, что продемонстрировано на рис. 88.

```
In[1]:= SphericalHarmonicY[2, 1, θ, φ]
Out[1]:=  $-\frac{1}{2} e^{i\phi} \sqrt{\frac{15}{2\pi}} \cos[\theta] \sin[\theta]$ 

In[2]:= TableForm[NSolve[SphericalHarmonicY[l, m, θ, φ] == 0 /. {l → 2, m → 1}, {θ, φ}]]
NSolve: Inverse functions are being used by NSolve, so some solutions may not be found; use Reduce for complete solution information.
NSolve: Equations may not give solutions for all "solve" variables.

Out[2]//TableForm=
θ → -1.5708
θ → 0.
θ → 1.5708

In[3]:= Reduce[SphericalHarmonicY[l, m, θ, φ] == 0 /. {l → 2, m → 1}, {θ, φ}]
Out[3]:=  $c_1 \in \mathbb{Z} \&\& \left( \theta == -\frac{\pi}{2} + 2\pi c_1 \parallel \theta == \frac{\pi}{2} + 2\pi c_1 \parallel \theta == 2\pi c_1 \parallel \theta == \pi + 2\pi c_1 \right)$ 
```

Рис. 88. Нули угловой части волновой функции электрона

Решение систем линейных алгебраических уравнений в матричной форме $a.x = b$, где a и b — известные матрица и вектор, относительно вектора x , выполняется командой *LinearSolve[a, b]*.

Пример 54. Найдем токи в электрической цепи на рис. 89, где ЭДС активных элементов равны $E_1 = 12$ В, $E_2 = 15$ В, $E_3 = 27$ В, а сопротивления резисторов $R_1 = 4$ Ом, $R_2 = 9$ Ом, $R_3 = 7$ Ом, $R_4 = 3$ Ом, $R_5 = 5$ Ом, $R_6 = 2$ Ом. Используя I и II законы Кирхгофа, запишем систему уравнений: $I_1 + I_2 - I_3 = 0$, $I_3 - I_4 - I_5 = 0$, $-I_1 + I_4 + I_6 = 0$, $R_1 I_1 + R_3 I_3 + R_4 I_4 = -E_1 + E_3$, $R_2 I_2 + R_3 I_3 + R_5 I_5 = -E_2 + E_3$, $R_4 I_4 - R_5 I_5 - R_6 I_6 = 0$. Решим ее относительно I_i , $i = 1, \dots, 5$, с помощью кода на рис. 90.

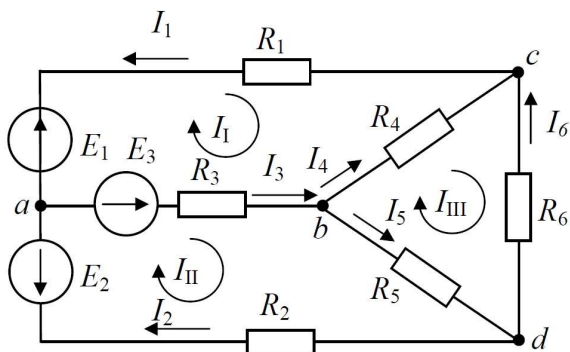


Рис. 89. Схема электрической цепи в примере 54

```

In[1]:= a = {{1, 1, -1, 0, 0, 0},
             {0, 0, 1, -1, -1, 0},
             {-1, 0, 0, 1, 0, 1},
             {r1, 0, r3, r4, 0, 0},
             {0, r2, r3, 0, r5, 0},
             {0, 0, 0, r4, -r5, -r6}};
MatrixForm[a]

Out[2]//MatrixForm=

$$\begin{pmatrix} 1 & 1 & -1 & 0 & 0 & 0 \\ 0 & 0 & 1 & -1 & -1 & 0 \\ -1 & 0 & 0 & 1 & 0 & 1 \\ r1 & 0 & r3 & r4 & 0 & 0 \\ 0 & r2 & r3 & 0 & r5 & 0 \\ 0 & 0 & 0 & r4 & -r5 & -r6 \end{pmatrix}$$


In[3]:= b = {0, 0, 0, -e1 + e3, -e2 + e3, 0}
Out[3]:= {0, 0, 0, -e1 + e3, -e2 + e3, 0}

In[4]:= LinearSolve[a, b] /.
{e1 -> 12., e2 -> 15, e3 -> 27,
 r1 -> 4, r2 -> 9, r3 -> 7, r4 -> 3, r5 -> 5, r6 -> 2}

Out[4]:= {1.03175, 0.174603, 1.20635, 0.809524, 0.396825, 0.222222}

```

Рис. 90. Решение системы уравнений Кирхгофа

11 Элементы математического анализа

11.1 Вычисление производных

Для дифференцирования с помощью Wolfram Mathematica существуют функции Dt и D , варианты использования которых приведены в табл. 28.

Таблица 28. Команды дифференцирования

$D[f, \{x, n\}, \{y, m\}, \dots]$	Вычисляет частную производную высших порядков $(\partial^m/\partial y^m)(\partial^n/\partial x^n)f$
$D[f, \{x, n\}]$	Вычисляет частную производную высших порядков $(\partial^n/\partial x^n)f$
$D[f, x, y, \dots]$	Вычисляет частную производную $(\partial/\partial y)(\partial/\partial x)f$
$D[f, x]$	Вычисляет частную производную $(\partial f/\partial x)f$
$Dt[f, \{x, n\}]$	Вычисляет полную производную n -го порядка $d^n f/dx^n$
$Dt[f, x]$	Вычисляет полную производную df/dx
$Dt[f]$	Вычисляет полный дифференциал df
$Dt[f, x_1, x_2, \dots]$	Вычисляет полную производную $d/dx_1 d/dx_2 \dots f$

При этом любой символ, которому ранее не было задано некоторое выражение, зависящее от параметров дифференцирования, функция D рассматривает как константу. В отличие от D , функция Dt считает все выражения, не содержащие символов, по которым производится дифференцирование, неявно зависящими от параметров дифференцирования. У нее существует опция *Constants*, пустая по умолчанию, которая указывает, какие переменные считать не зависящими от параметров дифференцирования.

Пример 55. Рассмотрим систему, состоящую из двух грузов с массами m_1 и m_2 , висающих на концах тонкой нерастяжимой нити длиной l , перекинутой через блок пренебрежимо малой массы (машина Атвуда). Из курса теоретической механики известно, что функцию Лагранжа данной системы можно записать в виде $L = \frac{m_1 + m_2}{2} \dot{z}^2 + g(m_1 - m_2)z + gm_2l$, где z — относительная координата. Составим уравнение Лагранжа 2-го рода $\frac{d}{dt} \frac{\partial L}{\partial \dot{z}} - \frac{\partial L}{\partial z} = 0$ и решим его с начальными условиями $\dot{z}(0) = v_{0z}$, $z(0) = z_0$, как продемонстрировано на рис. 91.

```
In[1]:= lagr = (m1 + m2)/2 * D[z[t], t]^2 + g*(m1 - m2)*z[t] + g*m2*l
Out[1]= g l m2 + g (m1 - m2) z[t] + 1/2 (m1 + m2) z'[t]^2

In[2]:= eqlagr =
Dt[D[lagr, D[z[t], t]], t, Constants -> {m1, m2, g, l}] - D[lagr, z[t]] == 0
Out[2]= -g (m1 - m2) + (m1 + m2) z''[t] == 0

In[3]:= sol = DSolveValue[{eqlagr, {D[z[t], t] /. t -> 0} == v0z,
z[0] == z0}, z[t], t] // Simplify
Out[3]= g (m1 - m2) t^2 / (2 (m1 + m2)) + t v0z + z0
```

Рис. 91. Решение уравнения Лагранжа 2-го рода

Пример 56. Проверим теорему о независимости порядка дифференцирования в смешанной производной $(\partial^2/\partial y^2)(\partial^2/\partial x^2)f = (\partial^2/\partial x^2)(\partial^2/\partial y^2)f$ для всюду непрерывной функции $f(x, y) = \exp(-x^2 \sin y)$ с помощью кода на рис. 92.

```
In[1]:= func = Exp[-x^2 * Sin[y]]
Out[1]= e^(-x^2 Sin[y])

In[2]:= D[D[func, x], y] - D[D[func, y], x]
Out[2]= 0
```

Рис. 92. Расчеты со смешанными производными

11.2 Вычисление интегралов

Mathematica позволяет выполнять аналитическое интегрирование, включая вычисление кратных интегралов, в конечных и бесконечных пределах с помощью функции *Integrate*, см. табл. 29.

Таблица 29. Использование функции *Integrate*

$\text{Integrate}[f, x]$	Вычисляет неопределенный интеграл $\int f dx$
$\text{Integrate}[f, \{x, x_{\min}, x_{\max}\}]$	Вычисляет определенный интеграл $\int_{x_{\min}}^{x_{\max}} f dx$
$\text{Integrate}[f, \{x, x_{\min}, x_{\max}\}, \{y, y_{\min}, y_{\max}\}]$	Вычисляет определенный интеграл $\int_{x_{\min}}^{x_{\max}} dx \int_{y_{\min}}^{y_{\max}} dy \dots f$

Пример 57. Проверим справедливость закона Стефана-Больцмана $R_\gamma(T) = \sigma T^4$ проинтегрировав распределение энергии по спектру излучения АЧТ (формулу Планка) $B(\nu, T) = \frac{2h\nu^3}{c^2} \left(e^{\frac{h\nu}{kT}} - 1 \right)^{-1}$, $R_\gamma(T) = \pi \int_0^\infty B(\nu, T) d\nu$ на рис. 93.

```
In[1]:= FormulaData[{"PlanckRadiationLaw", "Frequency"}]

Out[1]:= L[\nu] == 
$$\frac{\left( 2 h/c^2 \right) \nu^3}{-1 + e^{\frac{1 h \nu}{k T}}}$$


In[2]:= k = Entity["PhysicalConstant", "BoltzmannConstant"]["Value"][[1]];
h = Entity["PhysicalConstant", "PlanckConstant"]["Value"][[1]];
c = Entity["PhysicalConstant", "SpeedOfLight"]["Value"][[1]];

In[5]:= Pi*Integrate[(2*h)/c^2*nu^3/(-1 + Exp[h/k*nu/T]), {nu, 0., Infinity},
Assumptions -> {T \in Reals, T > 0}]

Out[5]:= 5.67037 \times 10^{-8} T^4

In[6]:= sigma = N[
Entity["PhysicalConstant", "StefanBoltzmannConstant"]["Value"][[1]]

Out[6]:= 5.67037 \times 10^{-8}
```

Рис. 93. Проверка закона Стефана-Больцмана

Совпадение полученного множителя при T^4 со значением σ из базы данных Mathematica свидетельствует о правильности результатов.

Пример 58. На рис. 94 проверим условие нормировки радиальной части волновой функции электрона в атоме водорода

$$\int_0^{\infty} |R(r, \theta, \varphi)|^2 r^2 dr = 1.$$

```
In[1]:= F[n_, l_, x_] := x^l * Exp[-x/2] * LaguerreL[n - l - 1, 2 l + 1, x];
R[n_, l_, r_] := (2/(n*a))^(3/2) *
  Sqrt[Factorial[n - l - 1]/(2*n*Factorial[l])] *
  F[n, l, 2 r/(n a)];
Integrate[Abs[R[3, 2, r]]^2 * r^2, {r, 0, Infinity},
Assumptions -> {a ∈ Reals, a ≥ 0}]
Out[3]= 1
```

Рис. 94. Проверка условия нормировки для $R(r, \theta, \varphi)$

В системе Wolfram Mathematica предусмотрена также возможность расчета эллиптических интегралов первого, второго и третьего рода, интеграла ошибок и прочих специальных функций.

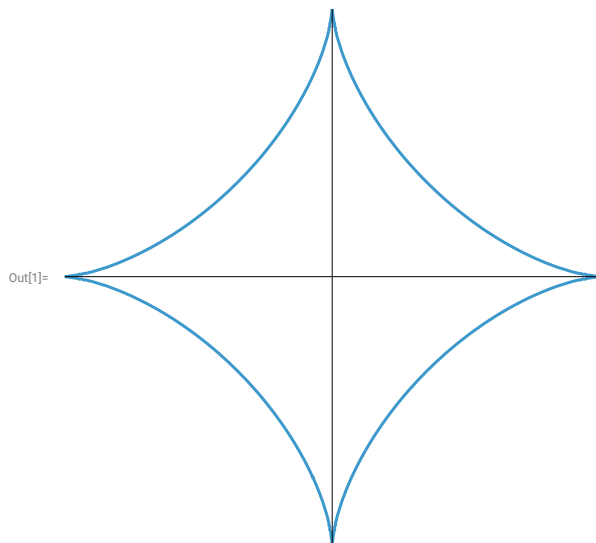
Пример 59. Рассчитаем период математического маятника длины l с максимальным углом отклонения θ_0 : $T = 4\sqrt{\frac{l}{g}} K(\sin^2 \frac{\theta_0}{2})$, см. рис. 95.

```
In[1]:= theta0 = Pi/3; l = 1; g = 9.81;
In[2]:= period = 4 * Sqrt[l/g] * EllipticK[Sin[theta0/2]^2]
Out[2]= 2.15287
```

Рис. 95. Расчет периода математического маятника

Пример 60. Известно, что длина дуги гладкой кривой $y = f(x)$ между точками $x = a$ и $x = b$ равна $l = \int_a^b \sqrt{1 + f'(x)^2} dx$. Рассчитаем длину дуги астроида $x^{2/3} + y^{2/3} = 1$, получившей свое название за схожесть графика со звездой, как показано на рис. 96.

```
In[1]:= ContourPlot[Surd[x^2, 3] + Surd[y^2, 3] == 1,
  {x, -1, 1}, {y, -1, 1}, PlotTheme -> "Minimal"]
```



```
In[2]:= lhs = x^(2/3) + y^(2/3)
```

```
Out[2]= x^(2/3) + y^(2/3)
```

```
In[3]:= eq1 = Dt[lhs == 1, x]
```

```
Out[3]= 2/(3 x^(1/3)) + 2 Dt[y, x]/(3 y^(1/3)) == 0
```

```
In[4]:= yprime = SolveValues[eq1, Dt[y, x]] /. 
```

```
Assumptions -> {(x^2)^(1/3) + (y[x]^2)^(1/3) == 1} //
```

```
Simplify // PowerExpand // Cancel
```

```
Out[4]= {-y^(1/3)/x^(1/3)}
```

```
In[5]:= squintegrand = Together[1 + yprime[[1]]^2] /. lhs -> 1
```

```
Out[5]= 1/x^(2/3)
```

```
In[6]:= Integrate[Sqrt[squintegrand], {x, 0, 1}]
```

```
Out[6]= 3/2
```

Рис. 96. Расчет длины дуги астроида

Определенные интегралы с численными пределами рассчитываются командой численного интегрирования *NIntegrate*. По умолчанию *NIntegrate* выполняет вычисления с машинной точностью, а желаемая точность результата контролируется опцией *PrecisionGoal*. Метод численного интегрирования выбирается автоматически, но его можно изменять с помощью опции *Method*.

Пример 61. Рассчитаем и изобразим на рисунке векторную диаграмму, иллюстрирующую разность фаз в случае дифракции Френеля на препятствии в виде полубесконечной непрозрачной плоскости с прямолинейным краем, которая называется спиралью Корню и задается в параметрическом виде с помощью интегралов Френеля $C(\omega) = \int_0^\omega \cos \frac{\pi \tau^2}{2} d\tau$ и $S(\omega) = \int_0^\omega \sin \frac{\pi \tau^2}{2} d\tau$, см. рис. 97.

```
In[1]:= ParametricPlot[{NIntegrate[Cos[Pi/2*t^2], {t, 0, ω}],  
NIntegrate[Sin[Pi/2*t^2], {t, 0, ω}]], {ω, -4., 4.},  
PlotStyle → Thick]
```

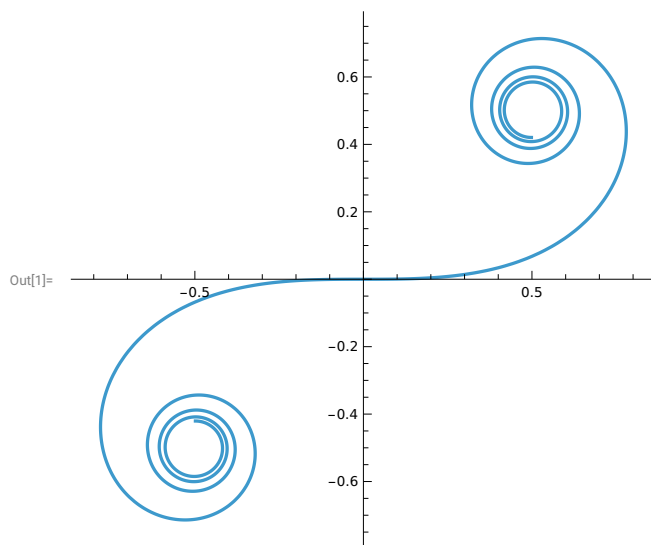


Рис. 97. Получение спирали Корню

11.3 Исследование функций на экстремум

В системе Wolfram Mathematica существует возможность нахождения локального минимума (максимума) явно заданной функции одной или нескольких переменных $f[x, y, \dots]$ при помощи встроенных функций $FindMinimum$ ($FindMaximum$) $[f, \{x, x_0\}, \{y, y_0\}, \dots]$, где x_0, y_0 — координаты стартовой точки процедуры поиска минимума (максимума).

Пример 62. Определим точки минимума и максимума калибровочно-инвариантного потенциала Хиггса в простой модели спонтанного нарушения симметрии $U(1)$, описываемого функцией $V(\phi) = -\mu^2 (\phi^\dagger \phi) + \lambda (\phi^\dagger \phi)^2 = \mu^2 |\phi|^2 + \lambda |\phi|^4$, где комплексное скалярное поле ϕ может быть записано в виде $\phi = |\phi| \exp(i\theta) = r \exp(i\theta)$. Сравним полученные значения с известным результатом при $\mu^2 > 0$, когда минимум потенциала находится в точке $|\phi| = v/\sqrt{2}$, где $v = (\mu^2/\lambda)^{1/2}$, см. рис. 98.

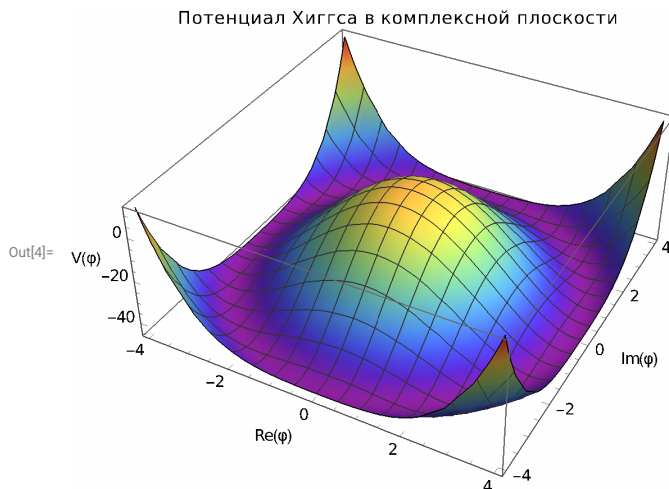
11.4 Интегральные преобразования

Пример 63. Пусть волновая функция квантовой частицы в координатном представлении имеет вид $\psi(x) = 1/\sqrt{2\pi} \exp(-x^2/2)$. Найдём для неё выражение в импульсном представлении применив к $\psi(x)$ преобразование Фурье, выражаемое формулой $\psi(p) = \frac{1}{2\pi} \int_{-\infty}^{\infty} f(x) e^{-ipx} dx$, и выведем график плотности вероятности обнаружить данную частицу с импульсом от -4 до 4. Также применим преобразование Меллина, выражаемое формулой $f(s) = \int_0^{\infty} f(x) x^{s-1} dx$ к $\psi(x)$ и выведем результат, как показано на рис. 99. Добавление слова *Inverse* в начало названия команд позволяет выполнить обратные преобразования, см. рис. 100.

```

In[1]:=  $\mu^2 = 6;$ 
 $\lambda = 0.2;$ 
 $V[x_, y_] := -\mu^2(x^2 + y^2) + \lambda(x^2 + y^2)^2$ 
Plot3D[V[x, y], {x, -4, 4}, {y, -4, 4},
  AxesLabel → {"Re( $\varphi$ )", "Im( $\varphi$ )", "V( $\varphi$ )"},
  PlotLabel → "Потенциал Хиггса в комплексной плоскости",
  ColorFunction → "Rainbow", Mesh → True, PlotRange → All]

```



```

In[5]:= Vr[r_,  $\theta$ ] := V[x, y] /. {y → r*Sin[ $\theta$ ], x → r*Cos[ $\theta$ ]} // Simplify
In[6]:= minResult = FindMinimum[Vr[r,  $\theta$ ], {r,  $\theta$ , 2}]
Out[6]:= {-45., {r → 3.87298}}
In[7]:= maxResult = FindMaximum[Vr[r,  $\theta$ ], {r,  $\theta$ , 2}]
Out[7]:= {0., {r → 0.}}
In[8]:= Sqrt[ $\mu^2/\lambda$  / 2]
Out[8]:= 3.87298

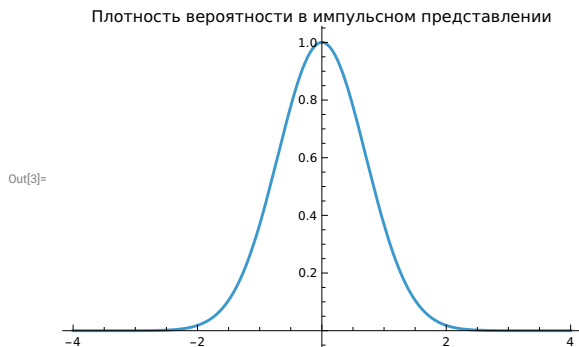
```

Рис. 98. Определение точек экстремума калибровочно-инвариантного потенциала Хиггса

```

In[1]:=  $\psi[x_] := 1/\text{Sqrt}[2*\text{Pi}]*\text{Exp}[-x^2/2]$ 
 $\psi[p_] := \text{FourierTransform}[\psi[x], x, p, \text{FourierParameters} \rightarrow \{1, -1\}]$ 
Plot[Abs[ $\psi[p]$ ]2, {p, -4, 4},
PlotLabel  $\rightarrow$  "Плотность вероятности в импульсном представлении"]

```



```

In[4]:=  $\text{MellinTransform}[\psi[x], x, u]$ 

```

Out[4]=
$$\frac{2^{-\frac{3}{2}+\frac{u}{2}} \text{Gamma}\left[\frac{u}{2}\right]}{\sqrt{\pi}}$$

Рис. 99. Преобразования Фурье и Меллина

```

In[1]:=  $\text{InverseFourierTransform}[\alpha \text{DiracDelta}[\omega] + \beta \text{DiracDelta}[\omega], \omega, t]$ 

```

Out[1]=
$$\frac{\alpha + \beta}{\sqrt{2\pi}}$$

```

In[2]:=  $\text{iSinc}[t_] = \text{InverseFourierTransform}[\text{Sinc}[\omega], \omega, t]$ 

```

Out[2]=
$$\frac{1}{2} \sqrt{\frac{\pi}{2}} (\text{Sign}[1 - t] + \text{Sign}[1 + t])$$

```

In[3]:= {Plot[Sinc[ $\omega$ ], { $\omega$ , -10, 10}],
Plot[iSinc[t], {t, -2, 2}, Exclusions  $\rightarrow$  None]}

```

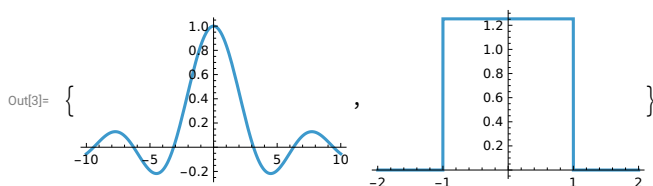


Рис. 100. Обратное преобразование Фурье

12 Дифференциальные уравнения

Решение большинства задач в физике сводится к решению соответствующих дифференциальных уравнений (ДУ). Пакет Wolfram Mathematica может находить решения обыкновенных дифференциальных уравнений, уравнений в частных производных и дифференциальных уравнений с запаздыванием.

12.1 Получение общего решения ДУ

Для получения общих решений дифференциальных уравнений в пакете Wolfram Mathematica существует функция *DSolve*, вызов которой может быть осуществлен в разных вариантах, см. табл. 30.

Таблица 30. Использование функции *DSolve*

<i>DSolve</i> [<i>eqn</i>]	Решает дифференциальное уравнение <i>eqn</i>
<i>DSolve</i> [<i>eqn</i> , <i>u</i> , <i>x</i>]	Решает дифференциальное уравнение <i>eqn</i> для функции <i>u</i> с независимой переменной <i>x</i>
<i>DSolve</i> [{ <i>eqn</i> ₁ , <i>eqn</i> ₂ , ...}, { <i>u</i> ₁ , <i>u</i> ₂ , ...}, ...]	Решает список дифференциальных уравнений <i>eqn</i> _{<i>i</i>}
<i>DSolve</i> [<i>eqn</i> , <i>u</i> , { <i>x</i> ₁ , <i>x</i> ₂ , ...}]	Решает дифференциальное уравнение <i>eqn</i> в частных производных

Аналогичный синтаксис имеет команда *DSolveValue*[*eqn*, *expr*, *x*], которая возвращает значение *expr*, определенное аналитическим решением обыкновенного дифференциального уравнения *eqn* с независимой переменной *x*.

Пример 64. Найдем изменение во времени количества ядер радиоактивных изотопов n_1 и n_2 из цепочки распада $n_1 \xrightarrow{\lambda_1} n_2 \xrightarrow{\lambda_1} n_3$. Решим систему уравнений, описывающую цепочку радиоактивного распада $\frac{dn_1(t)}{dt} = -\lambda_1 \cdot n_1(t)$, $\frac{dn_2(t)}{dt} = -\lambda_2 \cdot n_2(t) + \lambda_1 \cdot n_1(t)$, где $\lambda_{1,2}$ — постоянные распада ядер вида 1 и 2, $n_{1,2}(t)$ — изменение ко-

личества соответствующих радиоактивных ядер со временем, вначале последовательно, затем одновременно. Убедимся в идентичности полученных решений, см. рис. 101.

```

In[1]:= eqn1 = D[n1[t], t] == -λ1 * n1[t]
        eqn2 = D[n2[t], t] == -λ2 * n2[t] + λ1 * n1[t]

Out[1]:= n1'[t] == -λ1 n1[t]

Out[2]:= n2'[t] == λ1 n1[t] - λ2 n2[t]

In[3]:= sol1 = DSolve[eqn1, n1, t]
Out[3]:= {{n1 → Function[{t}, e-t λ1 c1]}}
```

```

In[4]:= eqn21 = eqn2 /. sol1
Out[4]:= {n2'[t] == e-t λ1 λ1 c1 - λ2 n2[t]}
```

```

In[5]:= sol2 = DSolveValue[eqn21, n2[t], t] // Simplify
Out[5]:= e-t λ2  $\left( \frac{e^{t(-λ1+λ2)} λ1 c_1}{-λ1 + λ2} + c_2 \right)$ 
```

```

In[6]:= sol0 = DSolve[{eqn1, eqn2}, {n1[t], n2[t]}, t]
Out[6]:= {{n1[t] → e-t λ1 c1, n2[t] → - $\frac{e^{-t λ1-t λ2} (-e^{t λ1} + e^{t λ2}) λ1 c_1}{λ1 - λ2} + e^{-t λ2} c_2$ }}
```

Рис. 101. Решение системы уравнений радиоактивного распада

Обратим внимание, что аналогично функции *Solve*, при записи уравнения в функции *DSolve* необходимо ставить операцию сравнения (==). Запись операции присваивания (=) приведет к ошибкам.

12.2 Решение уравнений с начальными условиями

Начальные условия задаются как дополнительные уравнения в первом аргументе функции *DSolve* или *DSolveValue*. Так как аргументов в данных функциях должно быть 3, то уравнение и начальные условия записываются в виде списка, т.е. в фигурных скобках.

Пример 65. Решим задачу о колебаниях двух математических маятников, связанных пружиной. Из курса механики известно, что данные колебания могут быть описаны системой дифференциальных уравнений $\ddot{x}_1 + \ddot{x}_2 + \omega_1^2(x_1 + x_2) = 0$, $\ddot{x}_1 - \ddot{x}_2 - \omega_2^2(x_1 - x_2) = 0$, где x_1, x_2 — отклонения маятников от положения равновесия, ω_1, ω_2 — числовые коэффициенты, зависящие от параметров маятников. Пусть в начальный момент времени $t = 0$ оба маятника покоятся, при этом один из маятников отклонен от положения равновесия на x_{10} . При данных начальных условиях решим систему уравнений с помощью функции *DSolve* и изобразим графики полученных решений при значениях $\omega_1 = 4.0, \omega_2 = 4.5, x_{10} = 1.$, см. рис. 102.

```
In[1]:= sol = DSolve[{D[x1[t], {t, 2}] + D[x2[t], {t, 2}] + w1^2*(x1[t] + x2[t]) == 0,
D[x1[t], {t, 2}] - D[x2[t], {t, 2}] + w2^2*(x1[t] - x2[t]) == 0,
x1[0] == x10, x2[0] == 0, x1'[0] == 0, x2'[0] == 0},
{x1[t], x2[t], t, Assumptions -> {x10, w1, w2} ∈ Reals}] // FullSimplify

Out[1]:= {{x1[t] -> 1/2 * x10 (Cos[t w1] + Cos[t w2]), x2[t] -> 1/2 * x10 (Cos[t w1] - Cos[t w2])}}
```

```
In[2]:= x1plot = x1[t] /. sol[[1]]; x2plot = x2[t] /. sol[[1]];
cond = {w1 -> 4, w2 -> 4.5, x10 -> 1};
Plot[{x1plot /. cond, x2plot /. cond}, {t, 0, 20},
PlotStyle -> {{Black, Dashed}, {Black, Thick}},
PlotLegends -> {"x1(t)", "x2(t)"}]
```

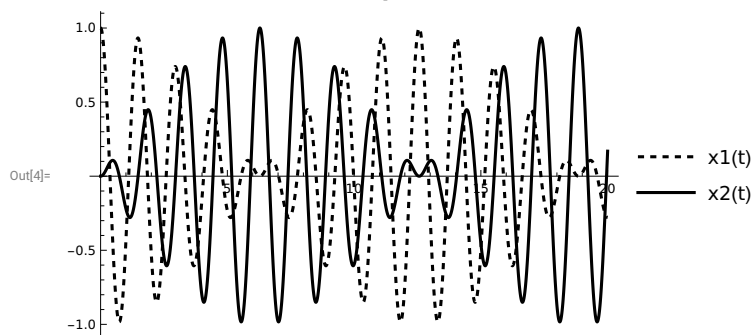


Рис. 102. Колебания связанных математических маятников

12.3 Численное решение дифференциальных уравнений

Для численного решения дифференциальных уравнений используется функция $NDSolve[lhs == rhs, y, \{x, x_{min}, x_{max}\}]$. Последний аргумент задается в виде списка и указывает, в каком диапазоне независимой переменной необходимо искать решение уравнения. Полученное решение выдается в виде интерполирующей функции $InterpolatingFunction[...]$, являющейся интерполяционным решением исходного дифференциального уравнения на заданном отрезке с указанными начальными условиями. Ее можно использовать как обычную функцию, например, для проверки начального условия или для графического представления решения. Отметим, что результат определен только в заданном при решении уравнения интервале. Для других значений результат хотя может и выводиться, но является неверным. Аналогичным образом можно решать системы уравнений.

Пример 66. Найдем решение нестационарного уравнения Шредингера, описывающего движение одномерного гармонического осциллятора $i\hbar \frac{\partial \Psi(x,t)}{\partial t} = -\frac{\hbar^2}{2m} \frac{\partial^2 \Psi(x,t)}{\partial x^2} + V(x)\Psi(x,t)$. Зададим $V(x) = \frac{1}{2}m\omega^2 x^2$ и выберем волновую функцию в виде $\psi_0(x) = (\frac{m\omega}{\pi\hbar})^{1/4} e^{-\frac{m\omega x^2}{2\hbar}}$ при $\hbar = m = \omega = 1$, см. рис. 103.

Пример 67. Рассмотрим движение точки, совершающей одновременно два гармонических колебания в двух взаимно перпендикулярных направлениях, описываемые системой уравнений $\ddot{x}(t) = -\omega_1^2 x(t)$, $\ddot{y}(t) = -\omega_2^2 y(t)$, где ω_1, ω_2 — частоты колебаний. В случае, если отношение ω_1/ω_2 является рациональным числом, совокупность положений точки в различные моменты времени будет представлять замкнутую кривую, называемую фигурой Лиссажу. Получим решение данной системы при $\omega_1/\omega_2 = 3/2$ и выведем на экран соответствующую фигуру Лиссажу. Для решения систе-

мы дифференциальных уравнений нужно заключить все уравнения и начальные условия в список, передаваемый команде *NDSolve* или *NDSolveValue*, обращая внимание, что разрывы строк не имеют никакого эффекта. После получения решения визуализируем его в виде параметрического графика, как показано на рис. 104.

```

In[1]:= hbar = 1;
        m = 1;
        ω = 1;

In[4]:= V[x_] := 1/2 * m * ω^2 * x^2
        eq =
        I * hbar * D[ψ[x, t], t] == -(hbar^2 / (2 * m)) D[ψ[x, t], {x, 2}] + V[x] * ψ[x, t];

In[6]:= inCond = ψ[x, 0] == Exp[-x^2 / 2.] / Sqrt[Sqrt[t] π];
        boundCond = {ψ[-5, t] == 0, ψ[5, t] == 0};

In[8]:= sol = NDSolve[{eq, inCond, boundCond}, ψ, {x, -5, 5}, {t, 0, 10}]

{{ψ → InterpolatingFunction[ Domain: {{-5., 5.}, {0., 10.}} Output: scalar ]}}

In[9]:= t1 = 4.;
        Plot[Evaluate[Abs[ψ[x, t1]]^2 /. sol], {x, -5, 5}, PlotRange → All,
        AxesLabel → {"x", "|ψ(x, t1)|^2"},
        Filling → Axis]

```

Out[10]=

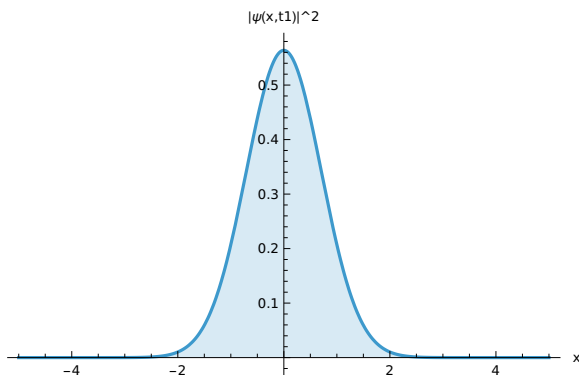


Рис. 103. Решение нестационарного уравнения Шредингера

```

In[1]:=  $\omega_1 = 3;$ 
 $\omega_2 = 2;$ 

In[3]:= eqns = {
  x'[t] == - $\omega_1^2$  * x[t],
  y'[t] == - $\omega_2^2$  * y[t],
  x[0] == 0., y[0] == 0.,
  x'[0] == 1., y'[0] == 2.
};

solutions = NDSolve[eqns, {x, y}, {t, 0, 2 Pi}]

Out[4]= {{x → InterpolatingFunction[
  Domain: {{0., 6.28}}
  Output: scalar
],
  y → InterpolatingFunction[
  Domain: {{0., 6.28}}
  Output: scalar
]}}

In[5]:= ParametricPlot[
  Evaluate[{x[t], y[t]} /. solutions],
  {t, 0, 2 Pi},
  AspectRatio → 1,
  AxesLabel → {"X", "Y"},
  PlotStyle → Thick, LabelStyle → Directive[Bold, Medium]
]

```

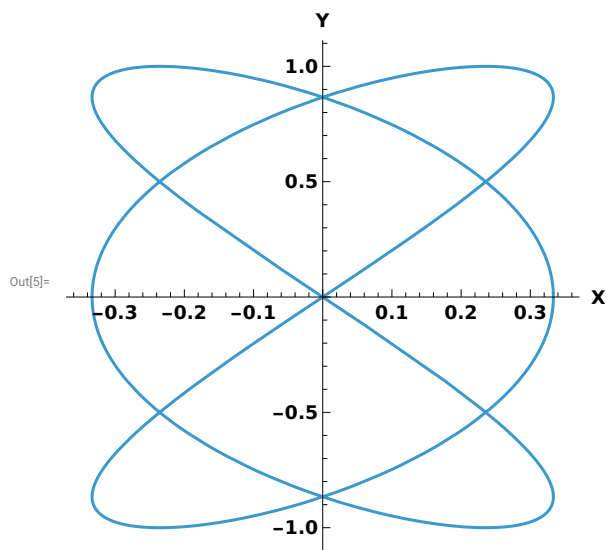


Рис. 104. Получение фигур Лиссажу

12.4 Собственные значения и собственные функции линейных дифференциальных операторов

Функция *DEigensystem* $[\mathcal{L}[u[x, y, \dots]], u, \{x, y, \dots\} \in \Omega, n]$ находит n первых (наименьших) СЗ и соответствующих им СФ линейного дифференциального оператора $\mathcal{L}$ в области Ω .

Пример 68. Найдем первые 6 собственных колебаний круглой мембраны, закрепленной на окружности единичного радиуса. Из математической физики известно, что координатная часть функции $U(t, x, y) = T(t)u(x, y)$, описывающей свободные колебания мембраны, подчиняется уравнению $\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} + \lambda u = 0$, эквивалентному уравнению на СЗ оператора Лапласа $-\Delta u = \lambda u$. Воспользуемся функцией *DEigensystem*, задав граничное условие $u(x, y) = 0$, как представлено на рис. 105.

```
In[1]:= {vals, funs} =
  DEigensystem[{-Laplacian[u[x, y], {x, y}],
    DirichletCondition[u[x, y] == 0, True]],
  u[x, y], {x, y} ∈ Disk[], 6];

In[2]:= vals
Out[2]:= {BesselJZero[0, 1]^2, BesselJZero[1, 1]^2, BesselJZero[1, 1]^2,
  BesselJZero[2, 1]^2, BesselJZero[2, 1]^2, BesselJZero[0, 2]^2}

In[3]:= funs
Out[3]:= {BesselJ[0, Sqrt[x^2 + y^2] BesselJZero[0, 1],
  x BesselJ[1, Sqrt[x^2 + y^2] BesselJZero[1, 1], y BesselJ[1, Sqrt[x^2 + y^2] BesselJZero[1, 1]]
  -----, -----,
  Sqrt[x^2 + y^2] Sqrt[x^2 + y^2]
  BesselJ[2, Sqrt[x^2 + y^2] BesselJZero[2, 1]] Cos[2 ArcTan[x, y]],
  BesselJ[2, Sqrt[x^2 + y^2] BesselJZero[2, 1]] Sin[2 ArcTan[x, y]],
  BesselJ[0, Sqrt[x^2 + y^2] BesselJZero[0, 2]]}

In[4]:= vals // N
Out[4]:= {5.78319, 14.682, 14.682, 26.3746, 26.3746, 30.4713}
```

Рис. 105. Решение задачи о колебаниях круглой мембраны

Визуализируем полученные решения с помощью функции

Plot3D, см. рис. 106.

```

In[5]:= Table[
  Plot3D[funs[[i]] // N // Evaluate, {x, y} ∈ Disk[],
  PlotRange → All, PlotLabel → vals[[i],
  PlotTheme → {"Minimal", "LargeLabels"}], {i, Length[vals]}]

```

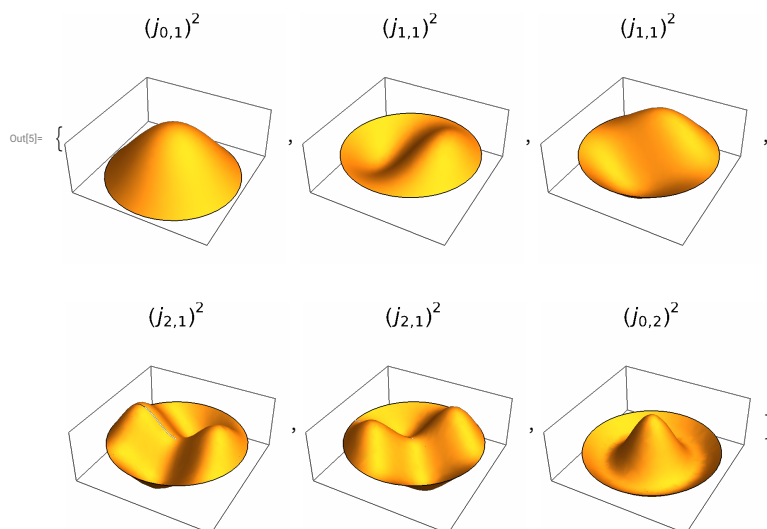


Рис. 106. Визуализация колебаний круглой мембраны

Вызов функции *DEigensystem* с параметрами *DEigensystem*[*eqns*, *u*, *t*, {*x*, *y*, ...} ∈ Ω, *n*] позволяет найти аналогичные СФ и СЗ зависящих от времени дифференциальных уравнений *eqns*. Если задача состоит в нахождении только СЗ, следует воспользоваться функцией *DEigenvalues* с идентичным синтаксисом вызова.

Для численного нахождения СЗ и СФ дифференциальных операторов используются аналогичные функции *NDEigenSystem* и *NDEigenValues*.

Пример 69. Решим задачу о нахождении СФ и СЗ обезразме-

ренного уравнения Шредингера $\frac{d^2\psi}{d\xi^2} + \xi^2\psi = \varepsilon\psi$, описывающего движение одномерного гармонического осциллятора в потенциале $V(x) = x^2$, как продемонстрировано на рис. 107.

```

In[1]:= {eig, func} =
NDEigensystem[-u''[x] + x^2*u[x], u[x], {x, -5, 5}, 10];

In[2]:= plotfunc = Flatten@Table[{eig[n], func[n]+eig[n]}, {n, 1, 10}];
colors =
Flatten@Table[{{RGBColor[0., 0.1*n, 0.2*n], RGBColor[0., 0.1*n, 0.2*n]}}, {n, 1, 10}];

In[4]:= Plot[x^2, plotfunc, {x, -4., 4.}, PlotStyle -> Join[Black, colors]]

```

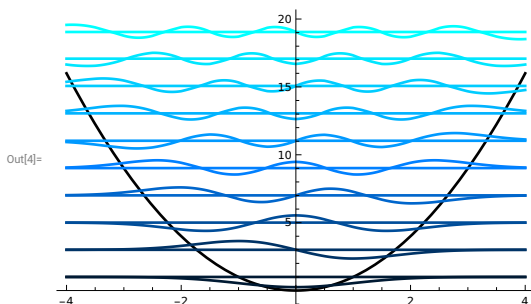


Рис. 107. Нахождение СФ и СЗ гармонического осциллятора

13 Графика

13.1 Графическое представление и аппроксимация данных

Одной из стандартных задач физики является поиск формулы, наилучшим образом описывающей заданный набор экспериментальных данных. Для решения этой задачи в Wolfram Mathematica предусмотрено несколько наиболее часто употребляемых функций, представленных в табл. 31.

Таблица 31. Команды поиска аппроксимирующей функции

$Fit[data, \{f_1, \dots, f_n\}, \{x_1, x_2, \dots, x_n\}]$	Находит линейную комбинацию функций f_i , зависящих от переменных $\{x_i\}$, наилучшим образом аппроксимирующую набор значений $data$
$FindFit[data, expr, pars, vars]$	Находит численные значения параметров $pars$, которые позволяют выражению $expr$ соответствовать данным $data$ наилучшим образом как функции $vars$.
$LinearModelFit[\{\{x_1, y_1\}, \{x_2, y_2\}, \dots\}, \{f_1, f_2, \dots\}, x]$	Строит линейную модель вида $a_0 + a_1 f_1 + a_2 f_2 + \dots$, которая аппроксимирует y_i для последовательных значений x_i
$NonLinearModelFit[\{\{x_1, y_1\}, \{x_2, y_2\}, \dots\}, form, \{a_1, \dots\}, x]$	Строит нелинейную модель по формуле $form$, которая аппроксимирует y_i для каждого из последовательных значений x_i используя свободные параметры a_i .

Пример 70. Рассмотрим материальную точку, движущуюся равноускоренно с постоянным ускорением $a = 2 \text{ м/с}^2$ и начальной скоростью $v_0 = 3 \text{ м/с}$ из точки с координатой $x_0 = 5 \text{ м}$. Будем замерять координату данной точки каждые 0.5 с за промежуток времени от 0 до 10 с с погрешностью измерения, не большей 4 м . Изобразим на рис. 108 зависимость измеренных координат точки от времени.

```
In[1]:= a = 2; v0 = 3; x0 = 5;
time = Range[0, 10, 0.5];
theorx = a*#^2 + v0*# + x0 & /@ time;
noise = 4.0;
expData =
Transpose[{time,
  theorx + RandomReal[{-noise, noise}, Length[time]]}];
expPlot = ListPlot[
  expData, PlotMarkers -> Automatic,
  AxesLabel -> {"Время (с)", "Расстояние (м)"}]
```

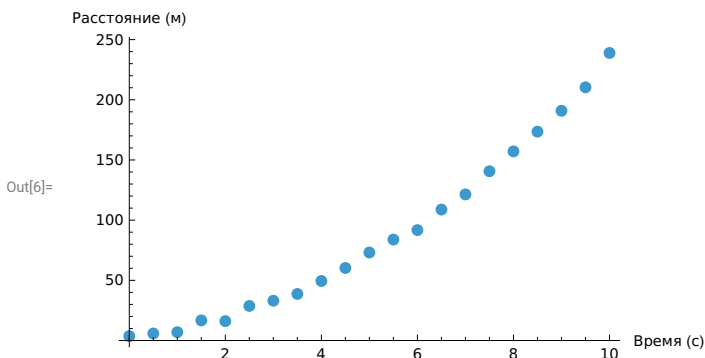


Рис. 108. Моделирование экспериментальных данных

Выполним аппроксимацию экспериментальных данных квадратичной функцией $x = a_0 + a_1 t + a_2 t^2$, определив ее параметры с помощью команды *Fit*. Сравним полученную зависимость с извест-

ной формулой для равноускоренного движения $x = x_0 + v_0 t + at^2/2$ и изобразим график полученной функции совместно с экспериментальными данными, как продемонстрировано на рис. 109.

```
In[7]:= f[x_] = Fit[expData, {1, x, x^2}, x]
```

```
Out[7]= 4.04776 + 3.52639 x + 1.94346 x^2
```

```
In[8]:= Show[expPlot, Plot[f[t], {t, 0, 10}]]
```

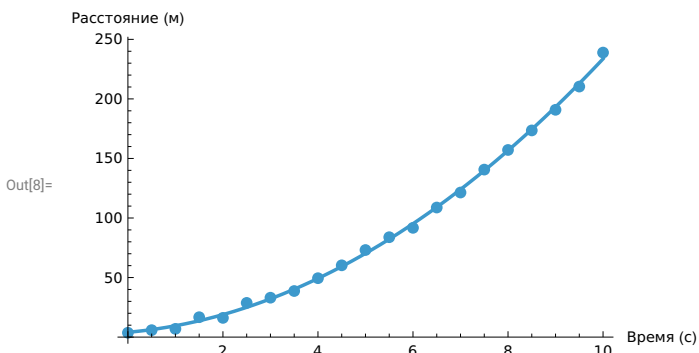


Рис. 109. Аппроксимация экспериментальных данных

Пример 71. Рассмотрим зависимость периода колебаний T математического маятника, движущегося в поле тяготения Земли, от его длины l . Сгенерируем набор экспериментальных данных с погрешностями *period* и выдвинем предположение о зависимости $T = a\sqrt{l}$. Для нахождения параметров аппроксимации заданной функцией воспользуемся командой *FindFit*. Представим экспериментальные данные совместно с графиком аппроксимирующей функции на рисунке, а также сравним полученный коэффициент пропорциональности a с известным значением $\frac{2\pi}{\sqrt{g}}$ с помощью кода на рис. 110. Обратим внимание на функцию *Around*, которая позволяет изобразить экспериментальные данные с учетом неопределенностей.

```

In[1]:= n = 20; g = 9.81;
errors = RandomReal[{0.1, 0.3}, n];
period[i_] := 2*Pi*Sqrt[(1 + 0.05)*i / g] + RandomReal[{-0.2, 0.2}];
data1 = Table[{(1 + 0.05)*i, Around[period[i], errors[[i]]], {i, 1, n}}];
Take[data1, 4]

Out[5]:= {{1.05, 2.25 ± 0.18}, {2.1, 3.05 ± 0.26}, {3.15, 3.62 ± 0.16}, {4.2, 4.12 ± 0.18}}

In[6]:= data2 = data1[[;;, 2, 1]];
Take[data2, 4]

Out[7]:= {2.25095, 3.05336, 3.62175, 4.11525}

In[8]:= model = a*Sqrt[x]
Out[8]:= a*Sqrt[x]

In[9]:= fit = FindFit[data2, model, {a}, x]
Out[9]:= {a -> 2.06946}

In[10]:= Show[Plot[Evaluate[model /. fit], {x, 0, 20}],
ListPlot[data1, AxesOrigin -> {0, 0},
AxesLabel -> {"l,m", "T,c"}, LabelStyle -> Medium]

```

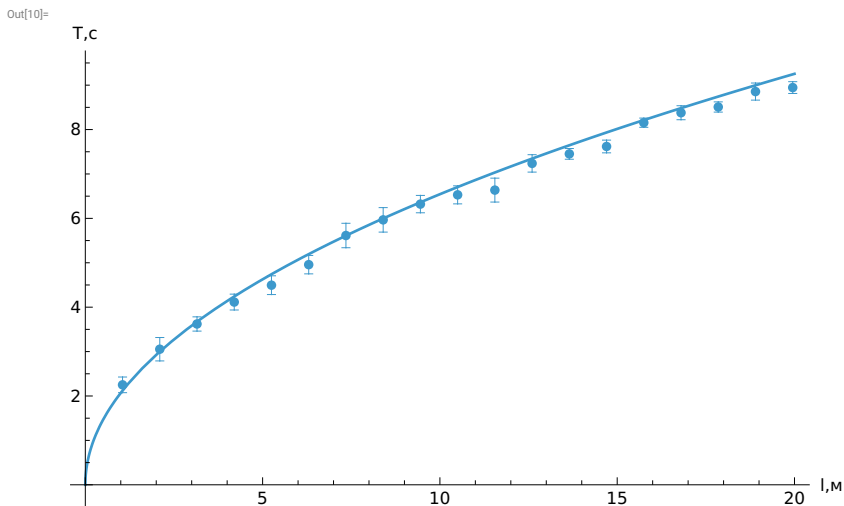


Рис. 110. Период колебаний T математического маятника

Пример 72. Рассмотрим распределение $N = 10000$ шариков в $rows = 10$ ячейках на доске Гальтона. Для этого воспользуемся встроенной функцией биномиального распределения

BinomialDistribution и представим результат в виде гистограммы, как представлено на рис. 111. Для построения двумерных гистограмм удобно использовать команды *Histogram* и *BarChart* в разных форматах и с широкими наборами опций. Запись указанных команд с добавлением *3D* в конце названия команды означает построение аналогичных трехмерных графиков, а добавление слова *List* в начало названия команды позволяет отображать списки точек с заданными координатами.

```
In[1]:= rows = 10; balls = 10 000;
ballDistribution =
Round[balls * PDF[BinomialDistribution[rows, 0.5], Range[0, rows]]];

In[3]:= hist = BarChart[ballDistribution, ChartLabels -> Automatic,
ChartStyle -> LightGray, ImageSize -> 400]
```

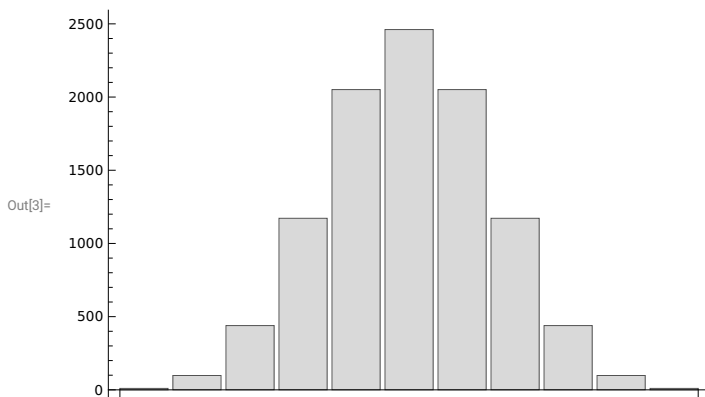


Рис. 111. Распределение шариков по ячейкам на доске Гальтона

В предположении о нормальном распределении шариков по ячейкам, для нахождения наилучших параметров, описывающих данное распределение, воспользуемся функцией *NonLinearModelFit*, после чего проиллюстрируем результаты на картинке, см. рисунок 112.

```

In[4]:= model = NonlinearModelFit[
  Transpose[{Range[1, rows + 1], ballDistribution}],
  a * PDF[NormalDistribution[μ, σ], x], {a, μ, σ}, x]

Out[4]= FittedModel[ 2.48 × 103 e-0.19 <<1>> ]

In[5]:= f[x_] = Normal[model]

Out[5]= 2476.09 e-0.190186 (-6.+x)2

In[6]:= Show[hist, Plot[f[x], {x, 1, rows + 1}, ImageSize → 400]]

```

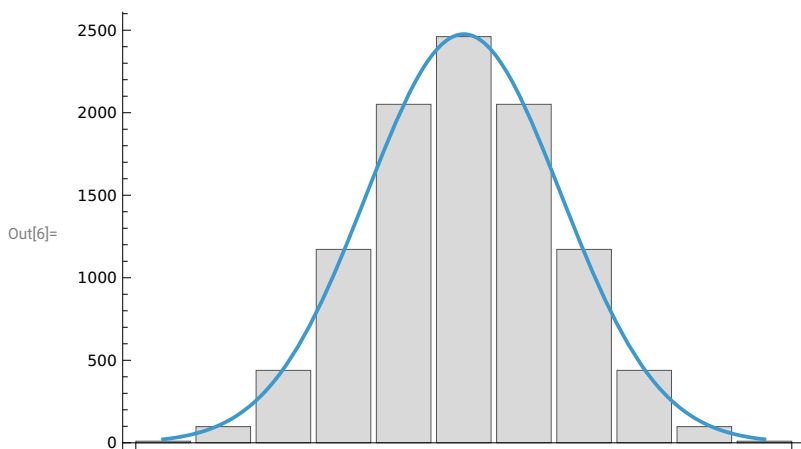


Рис. 112. Аппроксимирующая функция

13.2 Построение графиков

Команды для построения графиков в Wolfram Mathematica включают в себя слово *Plot*, а их применение порождает объект формата *Graphics* или *Graphics3D*. В простейшем случае команда $\text{Plot}[f[x], \{x, a, b\}]$ порождает график функции $f(x)$, когда x меняется от a до b . Всевозможные опции команды *Plot* вместе с их значениями по умолчанию могут быть получены командой $\text{Options}[\text{Plot}]$, часто используемые опции перечислены ниже.

1. Опция *AspectRatio* задает отношение высоты рисунка к его ширине, или форматное отношение. Форматное отношение по умолчанию равно $1/\textit{GoldenRatio}$ — пропорции рисунка соответствуют пропорциям листа формата A4 в альбомной ориентации. Задание $\textit{AspectRatio} \rightarrow 1$ означает, что рисунок вписывается в квадрат; $\textit{AspectRatio} \rightarrow \textit{Automatic}$ — система пытается по возможности задать одинаковый масштаб по осям.
2. *PlotRange* задает диапазон значений функции, которая на нем отображается. В случае выбора $\textit{PlotRange} \rightarrow \textit{All}$ система пытается поместить на график все значения функции f при x , меняющемся в заданных пределах, а $\textit{PlotRange} \rightarrow c, d$ предлагает команде строить только часть графика, лежащую в горизонтальной полосе $c \leq y \leq d$.
3. *PlotStyle* задает стиль графика, т.е. значения применяемых к нему графических директив. Типичными графическими директивами являются: ширина линий *Thickness* и абсолютная ширина линий *AbsoluteThickness*, штриховка *Dashing* и абсолютная штриховка *AbsoluteDashing*, описывающие плотность и прерывистость кривой; уровень серого *GrayLevel*, меняющийся от 0 (черный) до 1 (белый); директивы управления цветом: тон *Hue*, цвет *RGBColor*, задаваемый списком трех координат RGB (Red, Green, Blue) и др.
4. *AxesStyle* задает стиль координатных осей, например, $\textit{AxesStyle} \rightarrow \textit{Thick}$ строит оси жирными линиями.
5. *GridLines* задает координатную сетку, представляющую из себя прямые, параллельные координатным осям. Значение по умолчанию равно $\textit{GridLines} \rightarrow \textit{None}$, что означает отсутствие любых координатных прямых, кроме осей. Для выбора $\textit{GridLines} \rightarrow \textit{Automatic}$ прямые проводятся через метки (Ticks) на осях.

GridLines $\rightarrow x_1, \dots, x_m, y_1, \dots, y_n$ проводит прямые, параллельные оси y , через точки $x_1, \dots, x_m$ на оси x ; прямые, параллельные оси x — через точки $y_1, \dots, y_n$ на оси y .

6. *PlotLabel* задает заголовок или название графика, которое обычно пишется над графиком. Также часто используются следующие опции для создания подписей: *AxesLabel* — текст подписей к осям; *FrameLabel* — текст к обрамлению рисунка; *PlotLegends* — обозначения кривых на графике внутри рамки.

Включенный в задание опции (например, *PlotLabel*) текст нужно заключать в кавычки. Это превращает текст из последовательности символов в текстовый объект — строку (String), воспринимаемый и воспроизводимый буква в букву. Над строкой никакие обычные вычисления не производятся, для этого типа данных существуют специальные текстовые команды.

7. *Ticks* — метки или засечки, т.е. характерные точки, отмечаемые на осях. По умолчанию *Ticks* $\rightarrow$ *Automatic*, и система сама отмечает такие точки. Часто используются установки: *Ticks* $\rightarrow$ *None* — метки на осях не ставятся; *Ticks* $\rightarrow x_1, \dots, x_m, y_1, \dots, y_n$ — метки на оси x ставятся в точках $x_1, \dots, x_m$, а на оси y — в точках $y_1, \dots, y_n$.

8. *Filling* позволяет генерировать рисунки с заштрихованными областями под кривыми и поверхностями.

Существует возможность одновременно строить графики нескольких функций на одной картинке с помощью вызова команды *Plot* в формате *Plot*[{*f1*[*x*], *f2*[*x*], ...}, {*x*, *a*, *b*}]. Значения графических опций для списка функций также могут задаваться списком со взаимно однозначным соответствием его элементов элементам списка функций, как показано на рис. 113.

```

In[1]:= Plot[Evaluate@Table[BesselJ[n, x], {n, 3}], {x, 0., 10},
  AxesStyle -> {{Thick, Medium}, {Thick, Medium}},
  PlotLegends -> "Expressions",
  PlotStyle -> Evaluate@Table[{AbsoluteThickness[n],
  Dashing[{0.01*n, 0.02*n}], RGBColor[0., 0.3*n, 0.4*n]}, {n, 3}]

```

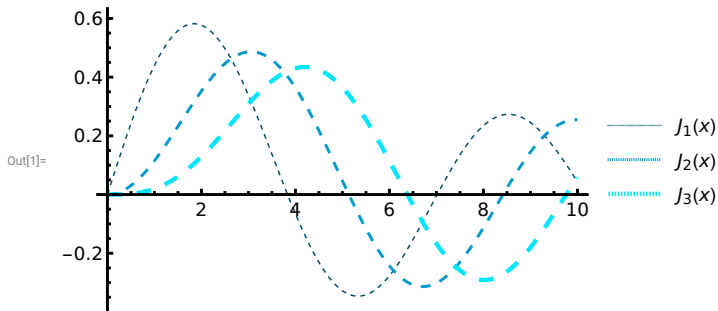


Рис. 113. Графики функций Бесселя

Значения графических опций, которые предполагается использовать для нескольких последующих графиков, можно объявить одной командой, предшествующей построению графиков, например: `SetOptions[Plot, AspectRatio -> 1, Ticks -> None]`. Указанные значения *AspectRatio* и *Ticks* будут действовать, пока не будут снова изменены или переопределены.

Для построения графиков специальных типов предусмотрены команды, в названиях которых слову *Plot* предшествует название типа графика. Например, `ParametricPlot[{f[t], g[t]}, {t, a, b}]` генерирует кривую, заданную параметрически $(f(t), g(t))$, когда t меняется от a до b . Возможный вариант использования команды *ParametricPlot* представлен в предыдущем разделе. Для построения изолиний (линий равных высот) используется команда *ContourPlot*. Изолинии — набор линий пересечения заданной поверхности с секущими горизонтальными плоскостями, расположенными параллельно друг другу, часто используемые для изображения

географических объектов и эквипотенциальных линий на поверхностях потенциальной энергии. Использование команды *ContourPlot* в формате *ContourPlot*[$f[x, y] == g[x, y], \{x, a, b\}, \{y, c, d\}]$ позволяет построить график неявно заданной функции. Построим семейство эллипсов с варьируемой большой полуосью, см. рис. 114.

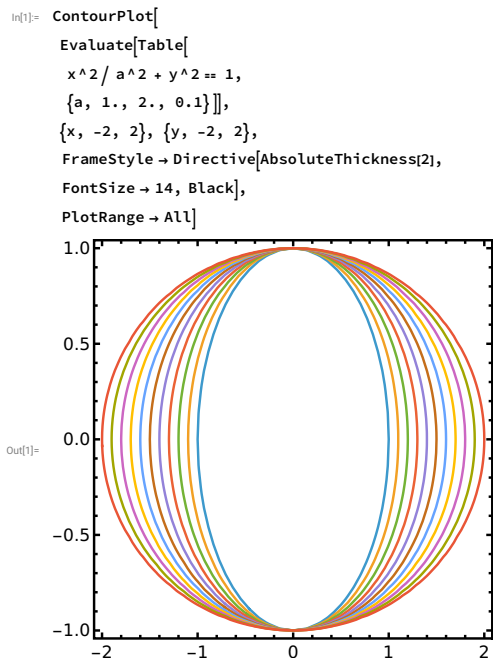


Рис. 114. Семейство эллипсов с варьируемой большой полуосью

PolarPlot[$\{r_1, r_2, \dots\}, \{\theta, \theta_{min}, \theta_{max}\}]$ генерирует график каждой зависимости r_i от θ на одном графике в полярных координатах.

Пример 73. Изобразим нормированные диаграммы направленности дипольной антенны с различными соотношениями длины антенны и длины волны передаваемого сигнала $n = l/\lambda$: $F(\theta) = \left(\frac{\cos(\frac{\pi l}{\lambda} \cos \theta) - \cos(\frac{\pi l}{\lambda})}{\sin \theta} \right)^2$ согласно рис. 115.

```
In[1]:= F = Table[((Cos[Pi*n*Cos[theta]] - Cos[Pi*n])/Sin[theta])^2, {n, 1, 4, 0.5}];
In[2]:= PolarPlot[F, {theta, -Pi + 0.01, Pi - 0.01}, PlotRange -> All, ImageSize -> Medium]
```

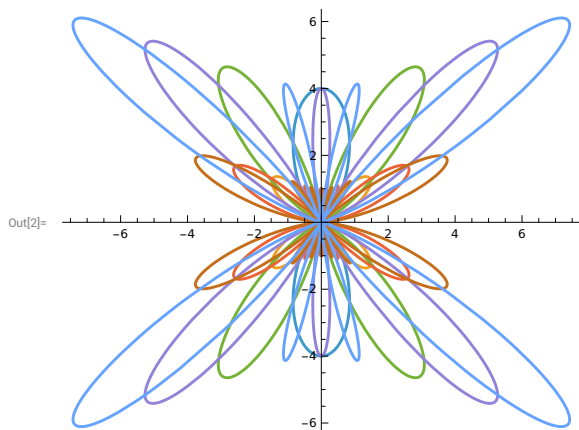


Рис. 115. Диаграммы направленности дипольной антенны

RevolutionPlot3D конструирует поверхность, образованную вращением кривой, заданной функцией f , вокруг оси z .

В системе Mathematica также предусмотрена возможность создания анимированных объектов. Анимация графики в реальном времени осуществляется функцией *Animate*, для интерактивного управления служит команда *Manipulate*.

Пример 74. Построим график плотности вероятности обнаружить квантовую частицу, находящуюся в основном энергетическом состоянии внутри бесконечно глубокой двумерной потенциальной ямы со сторонами a и b . Волновая функция такой частицы имеет вид $\psi(n_x, n_y, x, y) = \frac{2}{\sqrt{ab}} \sin(n_x \frac{\pi x}{a}) \sin(n_y \frac{\pi y}{b})$, как на рис. 116.

Поверхности в сферической системе координат строятся с помощью команды *SphericalPlot3D* $[\{r_1, r_2, \dots, r_n\}, \{\theta, \theta_{min}, \theta_{max}\}, \{\phi, \phi_{min}, \phi_{max}\}]$, которая генерирует трехмерный график с несколькими поверхностями.

```

In[1]:= a = 2; b = 2;
ψ[nx_, ny_][x_, y_] :=
  (2 / Sqrt[a*b]) * Sin[nx*Pi*x/a] * Sin[ny*Pi*y/b];
probDensity[nx_, ny_][x_, y_] :=
  Abs[ψ[nx, ny][x, y]]^2;
Plot3D[
  probDensity[1, 1][x, y],
  {x, 0, a},
  {y, 0, b},
  PlotRange → All,
  Mesh → Full,
  ColorFunction → "AtlanticColors",
  AxesLabel → {"x", "y", "P(x,y)"}
]

```

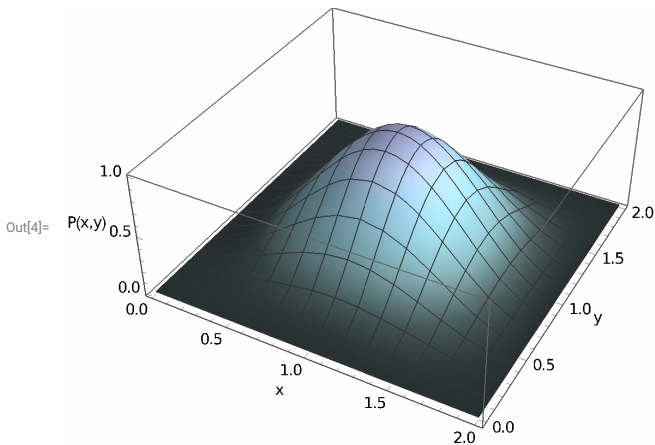


Рис. 116. Квантовая частица в двумерной потенциальной яме

Пример 75. Построим распределения угловой части плотности вероятности обнаружить электрон в атоме водорода в состояниях с $l = 2, m = 1$ и $l = 2, m = 2$ как показано на рис. 117.

Пример 76. Проиллюстрируем изменение формы потенциала Хиггса при изменении параметров μ^2 и λ на рис. 118.

```
In[1]:= SphericalPlot3D[{Abs[SphericalHarmonicY[2, 1,  $\theta$ ,  $\phi$ ]]^2,
Abs[SphericalHarmonicY[2, 2,  $\theta$ ,  $\phi$ ]]^2}, { $\theta$ , 0, Pi}, { $\phi$ , 0, 2*Pi},
PlotStyle -> Opacity[0.3], Mesh -> False]
```

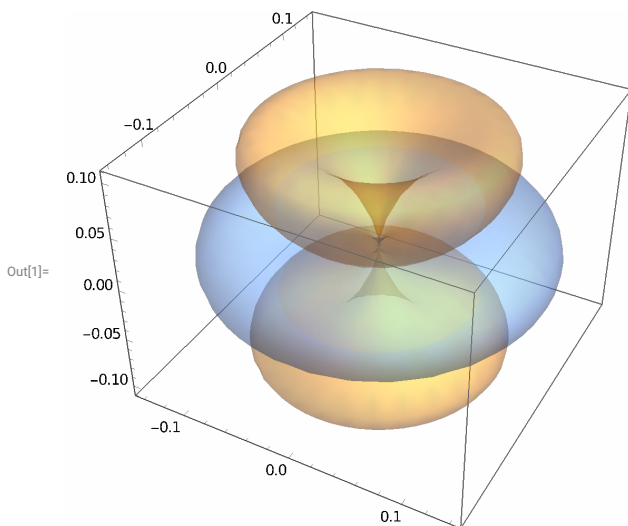


Рис. 117. Угловая часть плотности вероятности $Y_l^m(\theta, \varphi)$

```
In[1]:= V[r_,  $\mu_2$ _,  $\lambda$ _] := - $\mu_2$  * Abs[r * Exp[I *  $\theta$ ]]^2 +  $\lambda$  * Abs[r * Exp[I *  $\theta$ ]]^4 // ComplexExpand
In[2]:= Manipulate[RevolutionPlot3D[V[r,  $\mu_2$ ,  $\lambda$ ], {r, 0, 4}], { $\mu_2$ , 0, 6}, { $\lambda$ , 0, 2}]
```

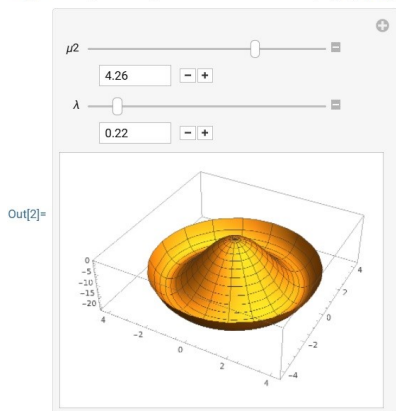


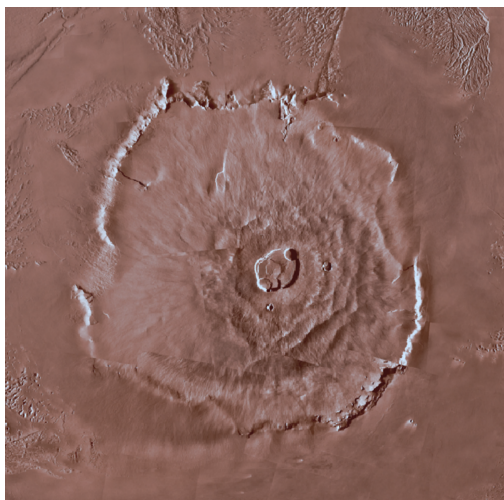
Рис. 118. Потенциал Хиггса в зависимости от μ^2 и λ

14 Импорт и экспорт

Для сохранения данных в файлы и их последующего считывания в Wolfram Mathematica предназначены команды *Import* и *Export*, а также различные их варианты. Они являются мощными инструментами для обмена данными между Wolfram Mathematica и внешними файлами в различных форматах: .png, .jpg, .csv, .txt, .xls, .mat, .wls, .mx и др. Команда *CloudExport*["filename", expr, "format"] сохраняет выражения, графики, таблицы, изображения и другие данные в файл *filename* в указанном формате *format*, который будет находиться в облачном хранилище среди других файлов пользователя.

Пример 77. Сохраним в формате .jpg карту поверхности Марса в радиусе 400 км вокруг горы Олимп, как показано на рис. 119.

```
In[1]:= graph1 = GeoGraphics[Entity["SolarSystemFeature", "OlympusMonsMars"],  
GeoRange -> Quantity[400, "Kilometers"]]
```



```
In[2]:= CloudExport[graph1, "JPG", "OlympusMonsMars.jpg"]
```

```
Out[2]:= CloudObject[https://www.wolframcloud.com/obj/entscheidend/OlympusMonsMars.jpg]
```

Рис. 119. Карта поверхности Марса вокруг горы Олимп

Команда `Import["filename", "format"]` считывает данные из файла *filename* и возвращает их в виде выражения Wolfram Mathematica.

Пример 78. Импортируем числовые данные из файла `numberdata.csv` в формате таблицы, а также данные о населении стран Земли из файла `population.xls`, находящиеся в папке со встроенными файлами примеров, как показано на рис. 120.

```
In[1]:= Import["ExampleData/numberdata.csv", "Table"]
```

```
Out[1]= {{1.2, 4.5, 6.7}, {5.4, 1.0, 0.0}, {0.0, 2.1, 3.1}}
```

```
In[2]:= Import["ExampleData/population.xls", "Dataset"]
```

Out[2]= {

1313973713.	China
1095351995.	India
298444215.	United States
245452739.	Indonesia
188078227.	Brazil
165803560.	Pakistan
147365352.	Bangladesh
142893540.	Russia
131859731.	Nigeria
127463611.	Japan

}

Рис. 120. Импорт данных

15 Элементы программирования

Стандартные операторы цикла и условные операторы приведены в табл. 32.

Таблица 32. Операторы цикла и условные операторы

<i>For</i> [<i>start</i> , <i>test</i> , <i>incr</i> , <i>body</i>]	Вычисляет выражение <i>start</i> , затем вычисляет <i>body</i> и <i>incr</i> до тех пор, пока <i>test</i> не примет значение <i>True</i>
<i>While</i> [<i>test</i> , <i>body</i>]	Вычисляет <i>test</i> , затем повторяет выполнение <i>body</i> до тех пор, пока <i>test</i> не потеряет значение <i>True</i> .
<i>Until</i> [<i>test</i> , <i>body</i>]	Вычисляет <i>body</i> и затем <i>test</i> повторяя до тех пор, пока <i>test</i> не примет значение <i>True</i> .
<i>If</i> [<i>condition</i> , <i>t</i> , <i>f</i>]	Выполняет <i>t</i> , если <i>condition</i> принимает значение <i>True</i> , и <i>f</i> , если значение <i>False</i>

Пример 79. Выполним моделирование 1000 шагов случайного блуждания молекулы внутри куба со стороной 10, где каждое новое положение молекулы генерируется случайным образом по каждой из трехмерных координат в пределах от -0.5 до 0.5. При этом, если предполагаемое новое положение молекулы оказывается вне куба, попытка отвергается, и новое положение вычисляется заново. Задать начальные данные как показано на рис. 121, дальнейший код представлен на рис. 122.

```
In[1]:= cubeSize = 10;  
pos = {RandomReal[{0, cubeSize}], RandomReal[{0, cubeSize}], RandomReal[{0, cubeSize}]};  
trajectory = {pos};  
maxSteps = 1000;  
stepSize = 0.5;
```

Рис. 121. Задание параметров моделирования

```

In[6]:= i = 0;
While[i < maxSteps,
  delta = RandomReal[{-stepSize, stepSize}, 3];
  newPos = pos + delta;
  isInside = True;
  For[j = 1, j ≤ 3, j++,
    If[newPos[j] < 0 || newPos[j] > cubeSize,
      isInside = False;
      Break[]; ]; ];
  If[isInside,
    pos = newPos;
    AppendTo[trajectory, pos]; i++;];
Graphics3D[{Blue, Tube[trajectory, 0.1],
  Red, Sphere[First[trajectory], 0.3],
  Green, Sphere[Last[trajectory], 0.3]},
  Axes → True, AxesLabel → {"X", "Y", "Z"},
  PlotRange → {{0, cubeSize}, {0, cubeSize}, {0, cubeSize}}]

```

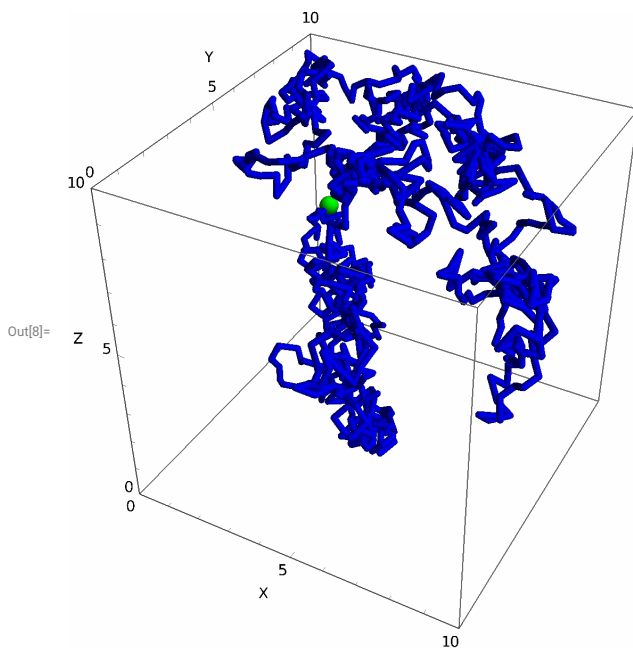


Рис. 122. Случайное блуждание молекулы в закрытом кубе

16 Полезные ресурсы

Практические руководства по использованию функционала Wolfram Mathematica на русском языке с решениями задач из разных разделов физики и математики представлены в учебных пособиях [12], [13] и [14].

Подробный учебник по основам компьютерной алгебры и языку Wolfram Language доступен на английском языке [15].

Wolfram Alpha [11] — вычислительная система знаний (computational knowledge engine), созданная компанией Wolfram Research, где ответы получаются путем выполнения динамических вычислений на основе обширной коллекции встроенных данных, алгоритмов и методов. Она ориентирована на ввод запросов в свободной форме и доступна для использования как в браузерной, так и в мобильной версии.

Wolfram Player [16] — бесплатное приложение, разработанное компанией Wolfram Research, которое позволяет просматривать и взаимодействовать с интерактивными документами, созданными в системе Wolfram. Может быть использован для просмотра интерактивных документов: файлов в форматах .nb, .cdf (Computable Document Format) и др. Открывает доступ к полному функционалу Wolfram Demonstrations Project [10].

Wolfram Community [17] — большое онлайн-сообщество пользователей технологий Wolfram для совместного решения проблем, обсуждения тематических вопросов и обмена идеями.

Список использованных источников

- [1] Matlab [Электронный ресурс]. – MathWorks, 2025. – URL: <https://www.mathworks.com/products/matlab.html> (дата обращения: 12.12.2025).
- [2] Statistica [Электронный ресурс]. – StatSoft, 2025. – URL: <https://www.statsoft.pl/en/software/statistica/> (дата обращения: 12.12.2025).
- [3] Mathcad [Электронный ресурс]. – PTC Inc, 2025. – URL: <https://www.mathcad.com/en> (дата обращения: 12.12.2025).
- [4] Wolfram Mathematica [Электронный ресурс]. – Wolfram, 2025. – URL: <https://www.wolfram.com/mathematica/?source=home> (дата обращения: 12.12.2025).
- [5] Maple [Электронный ресурс]. – Maplesoft, 2025. – URL: <https://www.maplesoft.com/products/Maple/> (дата обращения: 12.12.2025).
- [6] Maxima [Электронный ресурс]. – Maxima, 2025. – URL: <https://maxima.sourceforge.io/> (дата обращения: 12.12.2025).
- [7] Wolfram Language [Электронный ресурс]. – Wolfram, 2025. – URL: <https://www.wolfram.com/language/?source=home> (дата обращения: 12.12.2025).
- [8] FeynCalc [Электронный ресурс]. – Rolf Mertig, Frederik Orellana, Vladyslav Shtabovenko, 2025. – URL: <https://feyncalc.github.io/> (дата обращения: 12.12.2025).
- [9] Wolfram Cloud [Электронный ресурс]. – Wolfram, 2025. – URL: <https://www.wolframcloud.com/> (дата обращения: 12.12.2025).

- [10] Wolfram Demonstrations Project [Электронный ресурс]. – Wolfram Demonstrations Project & Contributors, 2025. – URL: <https://demonstrations.wolfram.com/> (дата обращения: 12.12.2025).
- [11] Wolfram Alpha [Электронный ресурс]. – Wolfram, 2025. – URL: <https://www.wolframalpha.com/?source=home> (дата обращения: 12.12.2025).
- [12] Салеев В. А. Система Mathematica для физиков: учебное пособие. Самара.: Изд-во Самарский университет, 2000. 140с.
- [13] Богданов А. В. Пакет Wolfram Mathematica для инженерных вычислений: учебное пособие. Томск.: Изд-во ТПУ, 2008. 104 с.
- [14] Вавилов Н. А., Халин В. Г., Юрков А. В. Wolfram Mathematica для нематематика: учебное пособие. М.: МЦНМО, 2021. 483 с.
- [15] Grozin A. Introduction to Wolfram Mathematica for Physicists. Springer. – 2014. 219p.
- [16] Wolfram Player [Электронный ресурс]. – Wolfram, 2025. – URL: <https://www.wolfram.com/player/> (дата обращения: 12.12.2025).
- [17] Wolfram Community [Электронный ресурс]. – Wolfram, 2025. – URL: <https://community.wolfram.com/> (дата обращения: 12.12.2025).

Учебное издание

Шипилова Александра Викторовна

**СИСТЕМА МАТЕМАТИКА
ДЛЯ МАТЕМАТИЧЕСКОГО МОДЕЛИРОВАНИЯ
В ФИЗИКЕ**

Учебное пособие

Редакционно-издательская обработка
издательства Самарского университета

Подписано в печать 21.04.2026. Формат 60×84 1/16.

Бумага офсетная. Печ. л. 7,75.

Тираж 27 экз. Заказ № . Арт. - 2(УП)/2026.

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«САМАРСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
УНИВЕРСИТЕТ ИМЕНИ АКАДЕМИКА С.П. КОРОЛЕВА»
(САМАРСКИЙ УНИВЕРСИТЕТ)
443086, САМАРА, МОСКОВСКОЕ ШОССЕ, 34.

Издательство Самарского университета
443086, Самара, Московское шоссе, 34.