Г. М. Фридман, С. Н. Леора

Математика & Mathematica

Избранные задачи для избранных студентов

Санкт-Петербург
2010

УДК 004.42::[512.64+517.1/.3]
ББК 32.973.26-018.2
Ф 88

Фридман Г. М., Леора С. Н.
Ф 88 Математика & Mathematica: Избранные задачи для избранных студентов. — СПб.: Невский Диалект; БХВ-Петербург, 2010. — 299 с.: ил.

ISBN 978-5-7940-0143-3

Предметом обсуждения является компьютерная математическая среда Mathematica (версия 7) и возможность ее практического использования для решения задач высшей математики в объеме первого курса физико-математических, инженерно-физических и экономико-математических специальностей высших учебных заведений.

В книге содержится большое количество решенных примеров из базовых учебников высшей школы. Вместе с тем она включает в себя ряд разработанных авторами модулей для решения типовых задач высшей и линейной алгебры, математического анализа и аналитической геометрии, поэтому будет интересна не только новичкам, но и продвинутым пользователям этого пакета.

Книга адресована прежде всего студентам. Она может быть полезна также аспирантам, инженерам, научным сотрудникам и преподавателям — всем, кто заинтересован в получении эффективного инструмента для выполнения всевозможных математических операций при помощи компьютера.

УДК 004.42::[512.64+517.1/.3]
ББК 32.973.26-018.2

ISBN 978-5-7940-0143-3

Оглавление

Введение

Эту книгу лучше читать, сидя за компьютером, на котором, естественно, установлена 7-я версия *Mathematica* (для большинства рассмотренных примеров годятся также и предыдущие версии программы). Авторы совершенно не претендуют на полноту изложения. Более того, весь текст — это лишь «интеллектуальный пинок» для новичков и не слишком опытных пользователей, дающий возможность быстро сделать первые самостоятельные шаги, набрать инерцию и двигаться дальше.

Работая с системой *Mathematica*, вы оказываетесь в насыщенной математической среде и приобретаете великолепные математические «мускулы». Как ими воспользоваться — вопрос, на который каждый отвечает сам. *Mathematica* — это айсберг возможностей, и каждый пользователь со временем формирует свою надводную часть.

Авторы на страницах этой книги делятся собственным многолетним опытом работы в этой мощной, внешне и внутренне привлекательной среде. Нельзя объять необъятное — банальная истина. В тексте содержится лишь часть идей, методов и готовых программных кодов из того, что хотелось бы донести до читателя, а значит, до потенциального пользователя этого замечательного пакета.

Мы хотим на примерах научить читателя пользоваться *Mathematica*, поэтому здесь не будет традиционного перечисления ее основных преимуществ, особенностей и возможностей. Мы будем обсуждать их по мере необходимости, когда это действительно понадобится для решения конкретных задач. Иногда будем без подробного объяснения приводить программный код, который можно скопировать — и пользоваться на здоровье.

Книга преимущественно посвящена решению задач по основным разделам высшей математики, которые в том или ином объеме проходят студенты в высшей школе на первом курсе. *Mathematica* дает конечный ответ — преподавателю же зачастую требуется развернутое решение. Поэтому, разбирая задачи, будем пытаться, по возможности, имитировать с помощью *Mathematica* то, что мы обычно делаем вручную, и затем проверять полученный результат, используя различные встроенные функции.

Надеемся, что эта книга будет полезна и студентам и преподавателям. Студенты получат хорошую поддержку и начнут решать задачи без ошибок. Преподаватели могут использовать возможности *Mathematica*, генерируя различные контрольные задания, причем сразу с известными правильными ответами, а еще лучше — включив ее в качестве компьютерного сопровождения в курс высшей математики.

1

Обзорная экскурсия

Язык определяет наш способ мышления
и определяет, о чем мы можем мыслить.
Б. Л. Ворф

Мы здесь не удержались и повторили эпиграф из замечательной книги Бьёрна Страуструпа «Язык программирования C++». Действительно, выбирая язык, мы выбираем среду, в которой будем работать и творить, и уже тем самым, конечно, в чем-то себя ограничим сообразно основополагающим принципам и парадигмам выбранного программного пакета. За все нужно платить, и поэтому, отдавая дань универсальности и высокой интеллектуальности данного средства, будем всегда помнить о подводных камнях, которые, несомненно, нас подстерегают.

Mathematica — интерактивная среда; вы находитесь в живом диалоге, можете по ходу исправлять свои ошибки, задавать вопросы. Потеряв нить диалога, можно вернуться к любому месту и начать все сначала. Тут главное не забывать, что *Mathematica* все помнит. Вы можете открывать новые файлы, закрывать старые, но все, что было выполнено во время текущего сеанса работы, она будет помнить. Например, в какой-то из программ было определено $x = 5$ и вы опять используете ту же переменную, уже забыв об этом присвоении, *Mathematica* честно везде подставит вместо x пятерку. Хорошо бы и на экзаменах так.

Mathematica — интеллектуальная среда, программа с характером, многое она делает «по умолчанию» сообразно своим, в общем-то разумным, законам. Нужно научиться подстраиваться под эти законы. Например, следует знать, что вычисляя предел функции, она всегда вычисляет предел справа, не предупреждая пользователя. И таких «умолчаний» у нее достаточно много.

Mathematica — это язык со своей грамматикой, как и любой другой язык. Программа, конечно, пытается скромно намекнуть пользователю о потенциальных ошибках (описках), чаще всего это касается синтаксиса. Если же она напрочь не понимает, что вы от нее хотите, или вы требуете от нее невозможного, она просто повторит ваш вопрос.

Mathematica — язык программирования, причем один из языков, который позволяет писать программный код, используя несколько различных стилей: традиционный процедурный, функциональный, основанный на создании правил преобразований.

Однако обо всем по-порядку...

1.1. Поехали!

Итак, мы запускаем *Mathematica*. Это обычная операция, ничего нового — либо дважды щелкнуть мышью по нужному ярлыку, либо в Start Programs (Пуск -> Все программы) найти нужную программу. Перед вами на экране — оболочка программы.

Вообще, *Mathematica* как программа состоит из двух основных частей: оболочки (так называемый `FrontEnd`) и ядра (*Mathematica* `Kernel`). Первая служит мостом между пользователем и системой: с ее помощью задаются вопросы и в удобной для пользователя форме формулируются ответы. Ядро — «рабочая лошадка» — ищет эти самые ответы, то есть выполняет все вычисления. Таков основной режим работы с системой: вопрос–ответ; в этом интерактивном режиме будем работать и мы. Итак, файл Untitled-1.nb к нашим услугам. В *Mathematica* рабочие файлы называются notebook и имеют соответствующее расширение nb.

Нажмем на клавишу `Enter`. Кое-что изменилось. Появилась синяя скобка в правом поле — она показывает, что открыта клетка ввода (input cell), в которую можно вводить один или несколько вопросов. Также появился мигающий вертикальный отрезок — положение курсора в клетке. Заметим, что до нажатия клавиши `Enter` на положение курсора указывала горизонтальная линия в самом верху рабочего поля. Попробуем задать первый сложный вопрос и наберем

```
1 + 1
```

Вопрос пока не задан; чтобы довести его до сведения *Mathematica* и получить ответ, следует нажать одновременно клавиши `Shift` и `Enter`, при этом курсор может находиться *в любом месте* клетки ввода. На наш взгляд, наиболее удобный способ — указательный или большой палец правой руки жмет на правый `Shift`, а потом средний палец той же руки жмет на `Enter`. Дело сделано! Смотрите, что происходит и что получается.

Как только первый вопрос задан, загружается ядро, поэтому и первый результат обычно приходится ждать немного дольше. Синяя скобка на это время зажглась приятным желтым цветом и стала двойной — вопрос обрабатывается, об этом же свидетельствует и надпись Running ... Untitled-1 в заголовке файла. Наконец работа окончена, мы видим

```
In[1]:= 1 + 1
Out[1]= 2
```

и понимаем, что

- главное — получился правильный ответ. *Mathematica* умеет складывать, а это уже немало;

- этот ответ появился в отдельной клетке вывода (output cell) со своей синей скобкой, причем скобка для вывода имеет другой, нежели скобка для ввода, контур;

- и вопрос, и ответ оказались пронумерованы: `In[1]` и `Out[1]`; более того, их клетки объединены специальной скобкой в общую клетку.

Вот, собственно, и вся *Mathematica*. Остальное — детали. Эти детали будем обсуждать в следующем режиме: клетка ввода с вопросами, клетка вывода с ответами, комментарии авторов.

1.2. Mathematica как БОЛЬШОЙ калькулятор

Рассмотрим некоторые возможности системы *Mathematica* в режиме «калькулятора», считающего намного быстрее и эффективнее обычных. Скажем больше — в режиме научного калькулятора, объединяющего в себе аналитические, численные, графические и иные возможности.

Начнем с арифметики. Запустим очередную клетку:

```
In[2]:= 12 - 5

Out[2]= 7
```

Это мы уже проходили: $12 - 5$ действительно равно 7. Как видите, нумерация вопросов и ответов продолжается. Нумерация сквозная и единая на все время сеанса работы с *Mathematica* (иногда это называют «сессией», не пугайтесь). Далее еще одно упражнение на арифметические действия:

```
In[3]:= 4 × 5 - 7 / 6 + 14^2 - 3 (11 + 3)
```

$$\text{Out[3]= } \frac{1037}{6}$$

Отметим, что операция умножения 4 на 5 обозначена специальным символом ×, который *Mathematica* сама поставит, как только мы введем пробел между этими числами. Для умножения годится и обычная звездочка `*`, и просто пробел между сомножителями. Если перемножаются число и переменная, то даже никаких дополнительных символов не нужно: `5a`. Однако нельзя при умножении писать без пробела `a5`, поскольку *Mathematica* воспримет эту конструкцию как имя новой переменной.

Приоритет выполнения арифметических операций в *Mathematica* стандартный: действия в круглых скобках, возведение в степень, умножение, сложение. Вопрос очередности выполнения тех или иных операций, естественно, очень важен. Каждому действию (не только арифметическому!) присвоен приоритет; это называется *precedence*, причем построена эта система так, чтобы, во-первых, максимально соответствовать принятым математическим стандартам, а во-вторых, чтобы уменьшить количество круглых скобок в выражениях. Например, приоритет операций сравнения ниже, чем у арифметических действий, поэтому можно вводить без скобок `a + b < c`.

Еще отметим, что ответ на предыдущий вопрос выдан в виде неправильной дроби (рационального числа). Дело в том, что это *точный* результат, так как в вопросе участвуют только целые, то есть точные, числа. Попробуйте, например, вместо 4 поставить `4.` и посмотрите, что получится [1]:

```
In[4]:= 4. × 5 - 7 / 6 + 14^2 - 3 (11 + 3)

Out[4]= 172.833
```

Согласитесь, в этом что-то есть: программа старается работать как можно лучше, обеспечивая максимально возможную для данного вопроса точность ответа.

Теперь немного больших чисел:

```
In[5]:= 2^1025

Out[5]= 359 538 626 972 463 181 545 861 038 157 804 946 723 595 \
  395 788 461 314 546 860 162 315 465 351 611 001 926 265 \
  416 954 644 815 072 042 240 227 759 742 786 715 317 579 \
  537 628 833 244 985 694 861 278 948 248 755 535 786 849 \
  730 970 552 604 439 202 492 188 238 906 165 904 170 011 \
  537 676 301 364 684 925 762 947 826 221 081 654 474 326 \
  701 021 369 172 596 479 894 491 876 959 432 609 670 712 \
  659 248 448 274 432
```

Mathematica будет считать *точно*, пока хватит ресурсов, прежде всего — памяти вашего компьютера. Для начинающих экстремалов заметим, что точный результат для $12^{128321825}$ на стареньком Intel Pentium 4, CPU 2.60 Hz с 1Gb RAM был найден примерно за 3 минуты. Авторы не осмелились взглянуть на сам ответ, поскольку цифр в нем больше чем атомов во вселенной. Часто в точной форме оценить или использовать полученный результат довольно трудно. Поэтому зададим такой вопрос:

```
N[%]
```

Эти загадочные символы означают следующее: применить к предыдущему ответу функцию с именем `N`. В системе *Mathematica* *очень* много (более тысячи) встроенных функций; с некоторыми мы познакомимся «по ходу пьесы».

Знак процента означает ссылку на предыдущий ответ, два знака `%%` — на предпоследний и т. д. Это экономит время и ресурсы. Однако если вы выполняли много действий в *Mathematica* и хотите использовать результат, найденный, скажем, 137 вопросов тому назад, то писать столько знаков процента, пожалуй, несолидно. Есть способ получше: чтобы использовать ответ на n-й по порядку вопрос (ввод) в сессии, достаточно набрать `Out[n]` либо `%n`. Поэтому раз предыдущий (последний по порядку) вывод имеет номер 5, то выше вместо знака `%` можно было бы написать `Out[5]` или `%5`:

`N[%5]` или `N[Out[5]]`

[1] Уверены, что здесь не имеет смысла обсуждать разницу в компьютерном представлении целых и вещественных чисел.

Итак, запустим клетку

In[6]:= **N[%]**

Out[6]= $3.595386269724632 \times 10^{308}$

Использованная здесь функция `N[expr]` выдает численное значение выражения `expr`. Имя функции — сокращение от numerical, то есть «численный».

Для записи всех функций *Mathematica* выработаны общие принципы: имя любой функции — английское слово, начинающееся с заглавной буквы и обозначающее то действие, которое должно быть выполнено (иногда это общепринятое сокращение); аргументы функции — в квадратных скобках и перечисляются через запятую. Последнее относится не только к встроенным, но и к определенным самим пользователем функциям. Такие принципы позволяют легко отличить, например, функцию `p[a + b]` от произведения `p(a + b)`.

При желании можно получить более подробную информацию о функции. В системе предусмотрена возможность быстрой подсказки (on-line help): нужно набрать в клетке ввода вопросительный знак и за ним имя функции без пробела, например `?N`, и запустить эту клетку на выполнение. Если полученных сведений вам недостаточно — наберите `??N`, тогда узнаете атрибуты данной функции и значения опций, установленные по умолчанию.

Функция `N` имеет и необязательный второй аргумент (он должен быть натуральным), который определяет количество точных знаков получаемого числа, как, например, это сделано в следующей клетке:

In[7]:= **N**$\left[\sqrt{\mathbf{10}}\,\mathbf{, 30}\right]$

N[π, 200]

Out[7]= 3.16227766016837933199889354443

Out[8]= 3.14159265358979323846264338327950288419716939937\
510582097494459230781640628620899862803482534211\
706798214808651328230664709384460955058223172535\
940812848111745028410270193852110555964462294895\
49303820

В одной клетке задано два вопроса — по вопросу на строку, каждый имеет свой порядковый номер.

Кроме того, стоит обратить внимание на то, что на клавиатуре нет символа π и знака квадратного корня. Их ввод осуществлен специальными ключами (набором клавиш), например π набрано как $\boxed{\texttt{Esc}}$`p`$\boxed{\texttt{Esc}}$ (`p` — первая буква в английском написании π; аналогично вводятся и остальные греческие буквы), а $\sqrt{10}$ — как $\boxed{\texttt{Ctrl+2}}$ 10 (договоримся, что знак «+», стоящий в рамке, означает *одновременное* нажатие двух и более клавиш).

Отметим, что константу π можно вводить и просто как `Pi`, что, конечно, менее красиво, но не менее надежно. Такой подход особенно должен нравиться тем, кто предпочитает работать с клавиатурой. Любители работать мышью могут для ввода специальных символов использовать так называемые палитры (palettes). Этих палитр в *Mathematica* несколько, но самую необходимую из них — `Basic Math Assistant` — вы, возможно, захотите держать под рукой. Добраться до нее очень легко через опцию меню `Palettes`. На этой же палитре размещены и встроенные константы, и различные другие структуры ввода.

О весьма широких возможностях математического набора в *Mathematica* поговорим в разделе 1.5, посвященном оболочке `FrontEnd`.

Следующая клетка показывает, что *Mathematica* без труда работает с комплексными числами:

```
In[9]:= (5 - 7 I)^25
Out[9]= 46 473 901 622 040 515 072 000 +
         227 247 671 433 189 562 093 568 ⅈ
```

Результат точный, поскольку и вопрос «точный»; мнимая единица вводится как `I` (а также `Esc` `ii` `Esc`); переход в регистр верхнего индекса для указания степени осуществлен набором `Ctrl+6`.

В следующей клетке — некоторые математические функции: гиперболический котангенс, экспонента, логарифм по основанию 2, косинус и синус (их аргументы задаются в радианах), функция Бесселя второго рода $Y_1(2)$, вещественная часть комплексной величины:

```
In[10]:= Re[Coth[1.5 I] -
           BesselY[1, 2] - Sin[π / 3] Log[2, √15] +
           Exp[2 I Cos[60 Degree]]
          ]
Out[10]= -1.0444
```

Если выражение содержит хотя бы одну величину с плавающей точкой (в данном примере — аргумент котангенса), то и результат будет выдан в том же формате. Сравните, например:

```
In[11]:= {Sin[2], Sin[2.], Sin[2] // N}
Out[11]= {Sin[2], 0.909297, 0.909297}
```

Заметим, что в предыдущих клетках нам несколько раз встречались так называемые *встроенные константы*, то есть символы, значение которых в *Mathematica* заранее определено. В частности, были использованы константы `Pi`, `I`, `Degree` (значение которой, как нетрудно догадаться, равно $\pi/180$), а также `E` (константа Непера, или основание натурального логарифма e, «красиво» вводится как `Esc` `ee` `Esc`) и многие другие, см. табл. 1. Имена встроенных констант защищены, их предусмотренные по умолчанию значения пользователь не сможет легко изменить.

Таблица 1

Некоторые встроенные константы системы *Mathematica*

«Простой» ввод	«Красивый» ввод	Комментарии
`Pi`	Esc p Esc	$\pi \approx 3.14159$
`I`	Esc ii Esc	$\mathrm{i}^2 = -1$
`E`	Esc ee Esc	$\mathrm{e} \approx 2.71828$
`Infinity`	Esc inf Esc	∞
`Degree`	Esc deg Esc	$\pi/180$
`GoldenRatio`		$(1+\sqrt{5})/2 \approx 1.61803$

Mathematica работает очень быстро, и зачастую результат, который вы получаете, недостижим в других аналогичных системах. Следующая клетка иллюстрирует скорость счета:

```
In[12]:= Timing[180 !]

Out[12]= {0.,
  200 896 062 499 134 299 656 951 336 898 466 838 917 540 \
   340 798 867 777 940 435 335 160 044 860 953 395 980 941 \
   180 138 112 097 309 735 631 594 101 037 399 609 671 032 \
   132 186 331 495 273 609 598 531 966 730 972 945 653 558 \
   819 806 475 064 353 856 858 157 445 040 809 209 560 358 \
   463 319 644 664 891 114 256 430 017 824 141 796 753 818 \
   192 338 642 302 693 327 818 731 986 039 603 200 000 000 \
   000 000 000 000 000 000 000 000 000 000 000 000}
```

Восклицательный знак, как обычно, означает факториал числа, то есть произведение всех натуральных чисел, его не превосходящих. Полное имя функции, как и следует ожидать, `Factorial[180]`. Функция `Timing` показывает и время в секундах, потраченное на поиск ответа, и сам ответ, объединяя их в фигурные скобки. Если вас интересует только время счета, то можете, например, поставить точку с запятой `;` после аргумента `Timing`. Этот символ весьма полезен, он «заглушает» любой вывод на экран:

```
In[13]:= Timing[5000 !;]

Out[13]= {1.33357 × 10^-17, Null}
```

Здесь `Null` и означает отсутствие вывода. Время, непосредственно потраченное на вычисление, впечатляет, но при этом, конечно, не стоит думать, что *Mathematica* умеет измерять столь малые временные интервалы. Просто все произошло очень быстро, быстрее чем за некоторый, наименьший для данной реализации системы, промежуток времени. Величина этого промежутка определена системной постоянной `$TimeUnit` и обычно равна одной миллисекунде.

Перейдем к более изощренным возможностям калькулятора под названием *Mathematica*. Для начала — немного пределов. «Предел» по-английски — limit, потому и имя функции записывается аналогично. Примеры взяты из задачника Б. П. Демидовича по математическому анализу [2]:

In[14]:= $\mathtt{Limit}\left[n^3\ \mathtt{ArcTan}\left[\frac{1}{n^3\left(x^2+1\right)+n\,x}\right]\mathtt{Tan}\left[\frac{\pi}{4}+\frac{x}{2\,n}\right]^n,\ n\to+\infty\right]$

Out[14]= $\frac{e^x}{1+x^2}$

In[15]:= $\mathtt{Limit}\left[\left(\frac{\mathtt{ArcSin}[x]}{x}\right)^{1/x^2},\ x\to 0\right]$

Out[15]= $e^{1/6}$

Комментировать тут почти нечего. Обратим только внимание на возможности математического набора; здесь формулы «двумерные», то есть имеют длину и высоту: числитель и знаменатель, степени и т. п. Если вы набираете формулы сами и еще не разобрались, как вводить такие объекты (на самом деле это совсем несложно, посмотрите в опции меню `Insert` опцию `Typesetting` или воспользуйтесь палитрами с помощью опции меню `Palettes`), то применяйте обычный одномерный ввод, например:

In[16]:= `Limit[(a^x - a^b) / (x - b), x → b]`

Out[16]= $a^b\ \mathtt{Log}[a]$

и при этом аккуратно расставляйте группирующие круглые скобки. Стрелка во втором аргументе функции `Limit` может быть введена как `->` (минус и знак «больше») или $\boxed{\mathtt{Esc}}$`->`$\boxed{\mathtt{Esc}}$.

Отметим, что *Mathematica* использует несколько различных видов скобок, например:

- круглые скобки ( ) применяются при формировании выражений;
- одиночные квадратные скобки [ ] используются для указания аргументов функций;
- фигурные скобки { } необходимы при формировании списков;
- двойные квадратные скобки [[ ]] применяются для обращения к определенным элементам списка (краткая форма записи функции `Part`);
- комментарии заключаются в скобки (* *);
- угловые скобки < > используются при «склеивании» строк (краткая форма записи функции `StringJoin`).

Следующая клетка еще раз иллюстрирует, что *Mathematica* может не только работать с числами, но и выполнять аналитические вычисления, к наиболее характерным из которых относятся дифференцирование и интегрирование.

```
In[17]:= D[Sin[ArcTan[x^2 + a]], x] // Simplify
         (* Simplify[D[Sin[ArcTan[x^2+a]],x]] *)
```

Out[17]= $$\frac{2\,x}{\left(1+\left(a+x^2\right)^2\right)^{3/2}}$$

Несложно догадаться, что функция `D`, имеющая два аргумента, дает возможность найти *частную* производную своего первого аргумента по переменной, указанной во втором аргументе.

Функция `Simplify`, стоящая после символов `//`, как следует из ее имени, упрощает полученное выражение. Отметим, что символы `//` указывают на то, что здесь использована так называемая *суффиксная* форма записи функции [1], когда сама функция пишется после своего аргумента. Также в клетке ввода показано, как выглядит *традиционная* форма записи, аналогичная суффиксной. Эта строка, следуя синтаксису *Mathematica*, *закомментирована* парами символов `(*` и `*)`. Упомянем еще одну форму записи функций — *префиксную*, включающую символ `@` и особенно удобную, когда используется цепочка функций (суперпозиция). Эквивалентность всех этих форм записи проиллюстрирована ниже:

```
In[18]:= Sin[x]
         Sin@x
         x // Sin

Out[18]= Sin[x]

Out[19]= Sin[x]

Out[20]= Sin[x]
```

Нужно обязательно помнить, что разные формы записи одной и той же функции имеют различный приоритет выполнения (precedence): у суффиксной, например, он весьма низкий, поэтому функция `f` в `expr//f` применяется *ко всему* выражению `expr`), а у префиксной значительно выше (даже выше, чем у операции возведения в степень), и это нужно учитывать. Попробуйте самостоятельно проанализировать следующие три клетки, в которых показано, в чем проявляется различие трех форм записи функций:

```
In[21]:= f[x + y]
         x + y // f

Out[21]= f[x + y]

Out[22]= f[x + y]
```

[1] В английском языке нет окончаний и суффикс стоит в конце слова.

```
In[23]:= f@x + y
         f@(x + y)

Out[23]= y + f[x]

Out[24]= f[x + y]

In[25]:= f@x^y
         f@(x^y)
         f@x^y

Out[25]= f[x]^y

Out[26]= f[x^y]

Out[27]= f[x^y]
```

Обратите также внимание на то, что *Mathematica* указывает номер только для первого вопроса, если в клетке ввода их несколько. При этом, однако, все ответы помещаются в отдельные клетки вывода, каждая из них пронумерована.

В следующем примере — третья производная от $\sin(x+a)\cos x$ по x. Переменная a рассматривается по умолчанию как параметр, от x не зависящий. Второй аргумент функции D имеет более сложную, но вполне очевидную структуру: переменная и количество дифференцирований:

```
In[28]:= D[Sin[x + a] Cos[x], {x, 3}]

Out[28]= -4 Cos[x] Cos[a + x] + 4 Sin[x] Sin[a + x]
```

На самом деле количество аргументов функции D, указывающих по какой переменной и сколько раз брать производную, не ограничено. Это особенно удобно, если вам нужно найти частную производную высшего порядка. Кроме того, и в первом аргументе можно указывать любое число дифференцируемых функций. Нужно только объединить их в знакомый нам уже *список*, то есть записать в фигурных скобках через запятую, например:

```
In[29]:= D[{Sin[x] Log[y] z, Sin[x y + z]},
          {x, 3}, z, {y, 2}]
```

Out[29]= $\left\{\dfrac{\text{Cos}[x]}{y^2},\ 6\,x\,y^2\,\text{Cos}[x\,y+z] + 6\,y\,\text{Sin}[x\,y+z] - x^2\,y^3\,\text{Sin}[x\,y+z]\right\}$

Ответ получен также в виде списка (массива) двух элементов. Если требуется найти полную производную функции или ее дифференциал, то используйте Dt, обобщающую действие D.

Дальше — больше. Интегрирование:

```
In[30]:= Integrate[
          Sin[x] Sin[2 x] Sin[3 x] Sin[4 x] Sin[5 x], x]
         ∫Sin[Log[x + a]] ⅆx
```

$$\text{Out[30]=} -\frac{\text{Cos}[x]}{16} - \frac{1}{48}\text{Cos}[3x] - \frac{1}{80}\text{Cos}[5x] + \frac{1}{176}\text{Cos}[11x] + \frac{1}{208}\text{Cos}[13x] - \frac{1}{240}\text{Cos}[15x]$$

$$\text{Out[31]=} \frac{1}{2}(a + x)\,(-\text{Cos}[\text{Log}[a + x]] + \text{Sin}[\text{Log}[a + x]])$$

Вообще, *интегратор*, ответственный в системе *Mathematica* за аналитическое интегрирование, предмет законной гордости разработчиков. Утверждается, что система способна найти практически все берущиеся неопределенные интегралы. Функция `Integrate` имеет очевидные по смыслу два аргумента. Отметим специальные символы во второй строке клетки ввода: знак интеграла, который вводится как Esc int Esc, а также дифференциал Esc dd Esc. В обычной одномерной записи задание выглядело бы так: `Integrate[Sin[Log[x+a]],x]`. Переменная `a` рассматривается как параметр.

Два более сложных неопределенных интеграла:

$$\text{In[32]:=} \int \frac{dx}{x\sqrt{x^4 - 2x^2 - 1}}$$

$$\text{Out[32]=} -\frac{1}{2}\,\text{ArcTan}\left[\frac{1 + x^2}{\sqrt{-1 - 2x^2 + x^4}}\right]$$

$$\text{In[33]:=} \int x\sqrt{1 + x^2}\,\text{Log}\left[\sqrt{x^2 - 1}\right] dx$$

$$\text{Out[33]=} \frac{1}{18}\left(12\sqrt{2}\,\text{ArcTanh}\left[\frac{\sqrt{1 + x^2}}{\sqrt{2}}\right] + \sqrt{1 + x^2}\left(-2\left(7 + x^2\right) + 3\left(1 + x^2\right)\text{Log}\left[-1 + x^2\right]\right)\right)$$

Следующий пример иллюстрирует работу с определенными интегралами, причем используется та же функция `Integrate`, что и для неопределенных:

```
In[34]:= Integrate[
          Sin[x] Sin[2 x] Sin[3 x] Sin[4 x] Sin[5 x],
          {x, 0, π/2}]
```

```
Out[34]= 64
         ---
         715
```

Задан «точный» вопрос, и ответ — рациональное число — также точен. Второй аргумент функции, заключенный в *фигурные* скобки, указывает как переменную, так и пределы интегрирования. *Mathematica* сначала находит первообразную, а потом подставляет заданные пределы, то есть работает по формуле Ньютона–Лейбница.

Первообразную можно найти не всегда, но сосчитать интеграл численно — другое дело:

```
In[35]:= NIntegrate[Sin[x] Sin[2 x]
           Sin[3 x] Sin[4 x] Sin[5 x], {x, 0 , π / 2}]

Out[35]= 0.0895105
```

Интересно, что соответствующая этой операции функция в *Mathematica* аналогична `Integrate`, но начинается с заглавной `N`, которая, напомним, представляет выражение в приближенном (численном) виде. В некотором смысле, ее действие можно описать как `N@Integrate`. Вообще в *Mathematica* довольно часто встречаются такие пары функций, одна из которых выполняет точное аналитическое действие, а вторая — приближенное численное, и к имени первой добавляется `N`.

По умолчанию в качестве метода интегрирования используются квадратуры Гаусса. Следует обратить особое внимание на то, чтобы подынтегральная функция при любом значении переменной интегрирования из указанного интервала приобретала бы числовое значение. В противном случае `NIntegrate` просто откажется работать, например:

```
In[36]:= NIntegrate[Sin[x], {x, 0, a}]
         NIntegrate[Sin[a x], {x, 0, π}]

         NIntegrate ::nlim :
           x = a is not a valid limit of integration . ≫

Out[36]= NIntegrate[Sin[x], {x, 0, a}]

         NIntegrate ::inumr :
           The integrand Sin[a x] has evaluated to
                 non−numerical values for all
                 sampling points in the region
                 with boundaries {{0, 3.14159 }}. ≫

Out[37]= NIntegrate[Sin[a x], {x, 0, π}]
```

В обоих случаях параметр `a` не определен (не имеет числового значения), при этом сначала «плохим» оказывается предел интегрирования, а затем — сама подынтегральная функция.

Для «двумерного» ввода определенных интегралов можно использовать и специальные ключи; например, такая последовательность клавиш

`Esc` int `Esc` `Ctrl+=` 0 `Ctrl+5` 1 `Ctrl+Space` x `Esc` dd `Esc` x

создаст в клетке ввода интеграл

$$\int_0^1 x \, dx$$

Это соответствует `Integrate[x,{x,0,1}]`. Заметим, что одновременное нажатие клавиши `Ctrl` и пробела, что обозначено выше как `Ctrl+Space`, возвращает курсор в базовое положение в строке.

Кратные (двойные, тройные и т. д.) интегралы также вполне по силам функции `Integrate`:

In[38]:= $\int_0^{\pi/4} \int_0^{4x} \mathrm{Sin}[x+y] \, dy \, dx$

Out[38]= $\frac{3\sqrt{2}}{5}$

Двумерный набор выполнен при помощи специальных ключей. В одномерной записи это выглядит так:

In[39]:= $\mathrm{Integrate}\left[\mathrm{Sin}[x+y], \left\{x, 0, \frac{\pi}{4}\right\}, \{y, 0, 4x\}\right]$

Out[39]= $\frac{3\sqrt{2}}{5}$

Важна последовательность указания переменных интегрирования в `Integrate`: сначала внешняя переменная, а потом внутренняя. Не перепутайте! *Mathematica* всегда делает только то, что попросите, так что пеняйте на себя.

Численное интегрирование в многомерном пространстве, как и в одномерном, выполняется функцией `NIntegrate`. Здесь правильное описание области интегрирования «жизненно» важно — если вы ошибетесь, то интегрируемая функция просто не получит требуемого значения (то есть она не будет принимать числовые значения) и, соответственно, появится уже встречавшееся нам сообщение об ошибке:

In[40]:= $\mathrm{NIntegrate}\left[\mathrm{Cos}[x+y], \left\{x, 0, \frac{\pi}{4}\right\}, \{y, 0, 4x\}\right]$

$\mathrm{NIntegrate}\left[\mathrm{Cos}[x+y], \{y, 0, 4x\}, \left\{x, 0, \frac{\pi}{4}\right\}\right]$

Out[40]= 0.0485281

NIntegrate ::nlim :

y = 4 x is not a valid limit of integration . ≫

Out[41]= $\text{NIntegrate}\left[\text{Cos}[x+y],\ \{y,0,4x\},\ \left\{x,0,\frac{\pi}{4}\right\}\right]$

Система *Mathematica* обладает огромными возможностями работы в режиме интерпретатора, выполняя разнообразные символьные операции, например разложение на простейшие сомножители:

In[42]:= $\text{Factor}\left[x^{65}+y^{65}\right]$

Out[42]=
$$(x+y)\left(x^4-x^3 y+x^2 y^2-x y^3+y^4\right)\left(x^{12}-x^{11} y+x^{10} y^2-x^9 y^3+x^8 y^4-x^7 y^5+x^6 y^6-x^5 y^7+x^4 y^8-x^3 y^9+x^2 y^{10}-x y^{11}+y^{12}\right)\left(x^{48}+x^{47} y-x^{43} y^5-x^{42} y^6+x^{38} y^{10}+x^{37} y^{11}-x^{35} y^{13}-x^{34} y^{14}-x^{33} y^{15}-x^{32} y^{16}+x^{30} y^{18}+x^{29} y^{19}+x^{28} y^{20}+x^{27} y^{21}-x^{25} y^{23}-x^{24} y^{24}-x^{23} y^{25}+x^{21} y^{27}+x^{20} y^{28}+x^{19} y^{29}+x^{18} y^{30}-x^{16} y^{32}-x^{15} y^{33}-x^{14} y^{34}-x^{13} y^{35}+x^{11} y^{37}+x^{10} y^{38}-x^6 y^{42}-x^5 y^{43}+x y^{47}+y^{48}\right)$$

Обратную операцию свертывания выражения в простую форму выполняет, в частности, уже встречавшаяся нам функция `Simplify`:

In[43]:= Simplify[%]

Out[43]= $x^{65}+y^{65}$

К тому же классу функций относятся `Expand`, `Apart`, `Together` и ряд других, позволяющих производить различные манипуляции с математическими выражениями, в том числе и с тригонометрическими (раскрывать скобки, приводить подобные члены, раскладывать на простейшие дроби и т. п.):

In[44]:= $\text{Apart}\left[\frac{x+1}{\left(x^2+1\right)(x-2)\,x^2}\right]$

Out[44]= $$\frac{3}{20\,(-2+x)}-\frac{1}{2\,x^2}-\frac{3}{4\,x}+\frac{1+3\,x}{5\left(1+x^2\right)}$$

In[45]:= $\text{Expand}\left[(a+b)^7\right]$

Out[45]= $$a^7+7 a^6 b+21 a^5 b^2+35 a^4 b^3+35 a^3 b^4+21 a^2 b^5+7 a b^6+b^7$$

```
In[46]:= TrigReduce[8 Sin[x] Sin[2 x] Sin[3 x] Sin[4 x] ]
Out[46]= 1 - Cos[6 x] - Cos[8 x] + Cos[10 x]
```

Функция FactorIntegrer раскладывает целое число на простые делители:

```
In[47]:= MyNumber = 81 765 593 871 073 292 309 348 345 125
         k = FactorInteger[MyNumber]
Out[47]= 81 765 593 871 073 292 309 348 345 125
Out[48]= {{5, 3}, {19, 2}, {1087, 1},
          {2 258 573, 1}, {4 410 613, 1}, {167 336 527, 1}}
```

Здесь переменной с именем MyNumber присвоено числовое значение, причем для этого использован знак =. Ответ *Mathematica* выдала в виде *списка* (list), каждый элемент которого — пара чисел (тоже список), причем первое из них — простой делитель числа, а второе — его степень. О том, как работать со списками, будет рассказано в разделе 1.3. Проверим, все ли в порядке:

```
In[49]:= 5^3 19^2 1087 × 2 258 573 × 4 410 613 × 167 336 527
         % == MyNumber
Out[49]= 81 765 593 871 073 292 309 348 345 125
Out[50]= True
```

Первая строка в клетке ввода понятна. Во второй — с помощью знака операции сравнения == мы сопоставляем переменную MyNumber с результатом умножения. Ответ True говорит о том, что левая и правая части совпадают.

Во избежание возможных недоразумений заметим, что *Mathematica* различает латиницу и кириллицу. И если где-то случайно вы вместо «икс» наберете «ха», то долго будете искать ошибку!

```
In[51]:= x = 4; х = 3;
         x + x
         х + х
         x =.; х =.;
Out[52]= 8
Out[53]= 6
```

Здесь «иксу» присвоено значение 4, а букве «ха» — 3. Внешне эти символы в стандартном шрифте *Mathematica* мало различаются, поэтому и такие странные, на первый взгляд, результаты. Операция присваивания выполнена с помощью знака =. Вообще, *Mathematica* пока что немного недолюбливает кириллицу, поэтому наш совет — просто не увлекаться ее использованием.

Займемся теперь решением различных уравнений и систем в *Mathematica*. Начнем с простых примеров, воспользовавшись услугами функции `Solve`:

In[55]:= $\mathbf{Solve}\left[x^3-6x+4==0,\ x\right]$

$\mathbf{Solve}\left[\sqrt[3]{2x-1}==x,\ x\right]$

$\mathbf{Solve}\left[\mathbf{Log}[x-a]==\mathbf{Log}\left[\sqrt{2x-1}\right],\ x\right]$

$\mathbf{Solve}\left[\left\{\sqrt{x}+y==2,\ x-y==1\right\},\ \{x,\ y\}\right]$

Out[55]= $\left\{\{x\to 2\},\ \left\{x\to -1-\sqrt{3}\right\},\ \left\{x\to -1+\sqrt{3}\right\}\right\}$

Out[56]= $\left\{\{x\to 1\},\ \left\{x\to \frac{1}{2}\left(-1+\sqrt{5}\right)\right\}\right\}$

Out[57]= $\left\{\left\{x\to 1-\sqrt{2}\ \sqrt{a}+a\right\},\ \left\{x\to 1+\sqrt{2}\ \sqrt{a}+a\right\}\right\}$

Out[58]= $\left\{\left\{x\to \frac{1}{2}\left(7-\sqrt{13}\right),\ y\to \frac{1}{2}\left(5-\sqrt{13}\right)\right\}\right\}$

Синтаксис совершенно прозрачен: сначала уравнение или система уравнений в фигурных скобках, а затем — неизвестные. Получены точные (аналитические) результаты, причем не только для полиномиальных уравнений. Ответы имеют специальный вид, это так называемые правила трансформации (transformation rules), при помощи которых, например, в любом выражении можно провести подстановку, заменяющую объект, расположенный в правиле слева от стрелки, на объект, стоящий справа. Сама подстановка осуществляется, в частности, функциями `Replace` или `ReplaceAll`. Последняя имеет удобную краткую форму записи `/.`:

In[59]:= $\mathbf{Solve}[\{x+2y==a,\ 2x+5y==b\},\ \{x,\ y\}]$

$\mathbf{Exp}[x]+\mathbf{Sin}[y]\ /.\ \%$

Out[59]= $\{\{x\to 5a-2b,\ y\to -2a+b\}\}$

Out[60]= $\left\{e^{5a-2b}-\mathrm{Sin}[2a-b]\right\}$

Конечно, `Solve` далеко не всегда находит все решения уравнения, о чем она честно предупреждает, выводя сообщение:

In[61]:= $\mathbf{Solve}[\mathbf{Sin}[x]==\mathbf{Sin}[2x],\ x]$

Solve::ifun :

Inverse functions are being used by Solve, so some solutions may not be found; use Reduce for complete solution information. ≫

Out[61]= $\left\{\{x\to 0\},\ \left\{x\to -\frac{\pi}{3}\right\},\ \left\{x\to \frac{\pi}{3}\right\}\right\}$

В таких случаях зачастую выручает более мощная функция Reduce с аналогичным синтаксисом:

```
In[62]:= Reduce[Sin[x] == Sin[2 x], x]
```

$$\text{Out[62]= } C[1] \in \text{Integers} \,\&\&\, \left(x == 2\pi C[1] \,||\, x == -\frac{\pi}{3} + 2\pi C[1] \,||\, x == \frac{\pi}{3} + 2\pi C[1] \,||\, x == \pi + 2\pi C[1]\right)$$

Функция NSolve *численно* определяет корни полиномиальных уравнений и их систем. По сути, ее работа эквивалентна применению функции N к результатам работы Solve:

$$\text{In[63]:= NSolve}\left[x^6 - x^5 + 2x^3 - 3 == 0,\ x\right]$$

```
Out[63]= {{x → -1.25187}, {x → -0.429579 - 0.852428 ⅈ},
  {x → -0.429579 + 0.852428 ⅈ},
  {x → 0.996562 - 1.16597 ⅈ},
  {x → 0.996562 + 1.16597 ⅈ}, {x → 1.11791}}
```

$$\text{In[64]:= NSolve}\left[\left\{2x^2 - 2xy + 5 == y,\ y^2 - 3 == x^2\right\},\ \{x, y\}\right]$$

```
Out[64]= {{x → 1.79029, y → 2.49101},
  {x → -0.0201446 - 1.75263 ⅈ,
   y → -0.120505 - 0.292983 ⅈ},
  {x → -0.0201446 + 1.75263 ⅈ,
   y → -0.120505 + 0.292983 ⅈ}}
```

Для более сложных уравнений, когда и подумать об аналитическом решении невозможно, годится функция FindRoot:

```
In[65]:= FindRoot[Cos[Sin[x]] == Exp[x^2 - 1], {x, 1}]

Out[65]= {x → 0.834305}
```

Она находит приближенное (численное) решение, иначе говоря, точку, в которой указанное равенство или система равенств выполняется с определенной, заранее заданной точностью. У функции два аргумента: первый — это то уравнение, корень которого мы ищем, в нашем случае это $\cos(\sin x) = \exp(x^2 - 1)$, а второй заключен в объединяющие фигурные скобки и указывает, что уравнение задано относительно переменной x, а также что точкой, с которой начинается поиск решения (начальным приближением), будет $x_0 = 1$. По умолчанию функция FindRoot реализует метод Ньютона с максимальным числом итераций 100. Поскольку уравнение нелинейное, оно может иметь несколько корней, но мы получаем только один из них, к которому сходятся итерации численного метода с выбранной начальной точки x_0. Если изменим x_0, то, возможно, найдем другое решение, например:

```
In[66]:= FindRoot[Cos[Sin[x]] == Exp[x^2 - 1], {x, -2}]

Out[66]= {x → -0.834305}
```

Может быть, процесс и не сойдется с заданной точностью за заданное по умолчанию число итераций, тогда *Mathematica* сообщит об этом на своем родном английском языке:

```
In[67]:= FindRoot[Cos[Sin[x]] == Exp[x^2 - 1], {x, -10}]

    FindRoot::cvmit :
      Failed to converge to the requested accuracy or
            precision within 100 iterations . ≫

Out[67]= {x → -1.1589}
```

Ответ, конечно, получен, но доверять ему не рекомендуется. Вот какие значения принимают левая и правая части уравнения в найденной точке:

```
In[68]:= {Cos[Sin[x]], Exp[x^2 - 1]} /. %

Out[68]= {0.608711, 1.40923}
```

Как видите, совсем не то. Увеличив допустимое число итераций с помощью так называемой *опции*, все же добьемся своего:

```
In[69]:= FindRoot[Cos[Sin[x]] == Exp[x^2 - 1],
           {x, -10}, MaxIterations → 150]
         {Cos[Sin[x]], Exp[x^2 - 1]} /. %

Out[69]= {x → -0.834305}

Out[70]= {0.737909, 0.737909}
```

Функция DSolve аналитически решает дифференциальные уравнения и их системы, используя различные встроенные алгоритмы, в том числе те, которые применены в Integrate и Solve. Сначала дадим переменной с именем eqn значение дифференциального уравнения, которое будем решать:

In[71]:= $\left(\texttt{eqn} = \texttt{y''[x]} - \dfrac{\texttt{y'[x]}}{\texttt{x}} == \texttt{x Sin[x]}\right)$ // TraditionalForm

Out[71]//TraditionalForm=

$$y''(x) - \frac{y'(x)}{x} = x\sin(x)$$

Штрих после y в клетке ввода обозначает производную функции одной переменной. Набирается этот штрих клавишей «кавычка». Полная запись (то есть FullForm, полное имя функции) для y′ имеет вид Derivative[1][y], а для y'', очевидно, Derivative[2][y].

На самом деле удобнее рассматривать Derivative как *оператор дифференцирования*, что иллюстрируется примером, в котором производная взята только от функции Sin:

In[72]:= Sin'[x + x^2]

Out[72]= Cos$\left[\texttt{x} + \texttt{x}^2\right]$

Функция `TraditionalForm` выводит выражение в традиционной математической нотации — это лишь определенный формат отображения результатов. Самостоятельно сравните вывод в традиционной и рабочей (обычной для *Mathematica*) записи, запустив клетку с `eqn`. Обратите внимание на то, что `TraditionalForm` была применена к переменной `eqn` *после того*, как эта переменная получила свое значение, и поэтому по сути не изменила его. В этом заключается принципиальное отличие от похожей записи, где переменная `eqn1` получает значение уже в формате `TraditionalForm`, хотя выводы в обоих случаях совершенно одинаковы:

```
In[73]:= eqn1 = y''[x] - y'[x]/x == x Sin[x] // TraditionalForm
```

Out[73]//TraditionalForm=

$$y''(x) - \frac{y'(x)}{x} = x\sin(x)$$

У `DSolve` три обязательных аргумента: дифференциальное уравнение, функция, которую мы ищем, и переменная, от которой зависит неизвестная функция:

```
In[74]:= DSolve[eqn, y, x]
```

```
Out[74]= {{y → Function[{x},
             1/2 x^2 C[1] + C[2] - Cos[x] - x Sin[x]]}}
```

Не будем вдаваться в детали, отметим только, что ответ получен в виде правила подстановки для искомой функции `y` да еще и как список списков (две пары фигурных скобок), при этом, само собой, появились две неопределенные константы интегрирования, обозначенные по умолчанию как `C[1]` и `C[2]`.

Следующую клетку ввода тоже оставим почти без комментариев. В ней мы применяем (в *Mathematica* это называется «подставляем») найденное решение при помощи операции `/.` (полное имя функции — `ReplaceAll`), даем ему имя `solution` и выводим его в традиционной математической нотации:

```
In[75]:= solution = First[y /. DSolve[eqn, y, x]];
         solution[x] // TraditionalForm
```

Out[76]//TraditionalForm=

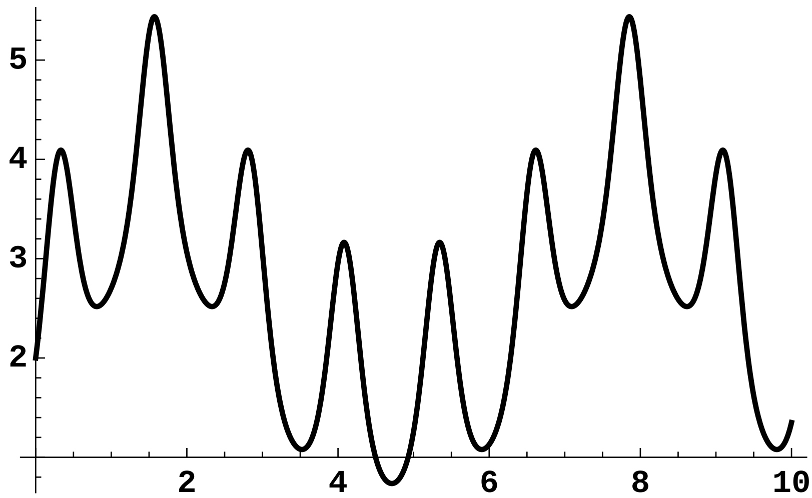
$$\frac{c_1 x^2}{2} + c_2 - x\sin(x) - \cos(x)$$

Раз есть `DSolve`, то где-то неподалеку есть и `NDSolve`, которая должна решать дифференциальные уравнения численно. Но об этом чуть позже, а пока немного порисуем.

Различные рисунки и графики в *Mathematica* легко генерировать с помощью так называемых графических функций `Plot`, `Plot3D` и т. д.:

```
In[77]:= Plot[Exp[Sin[x]] + Exp[Sin[5 x]],
          {x, 0, 10}, PlotStyle → {Black, Thick}]
```

Out[77]=



Здесь у `Plot` *три* аргумента. Первые два в точности повторяют аргументы функции `NIntegrate` и определяют, что́ рисуем и где (в каком промежутке) это делаем. При этом, конечно, указывается и переменная, от которой зависит изображаемая на графике функция. Третий аргумент с именем `PlotStyle` дает возможность «аранжировать» график, точнее, определять стиль рисуемой линии, то есть в данном случае цвет и толщину линий.

Как видите, мы опять столкнулись с аргументом функции, имеющим имя (впервые, напомним, мы встретили такой синтаксис на с. 23, когда меняли допустимое количество итераций для метода Ньютона в `FindRoot`). Вообще, все аргументы всех функций в системе *Mathematica* делятся на *позиционные* и *именные*[1] (positional and named arguments). Первые должны появляться в строго определенном месте, как, например, промежуток интегрирования в `NIntegrate` или переменная, по которой проводится дифференцирование в `D` (всегда и обязательно вторые аргументы) и т. п. Поэтому имена им ни к чему, *Mathematica* распознает их по месту расположения. Именные аргументы, естественно, опознаются по именам, и порядок, в каком они встречаются, неважен, лишь бы он не входил в противоречие с положением позиционных аргументов. Именные аргументы в *Mathematica* носят название *опции* и имеют общий синтаксис:

`option -> value`,   иногда   `option :> value`

где `option` — имя опции (аргумента), а `value` — ее значение. Опции являются необязательными аргументами, все они имеют некоторое заранее определенное по умолчанию значение (default value), которое и используется, если не изменено пользователем.

Для функции с именем `func` все ее опции и их значения, принятые по умолчанию, можно посмотреть при помощи `Options[func]`. Поскольку у `Plot` опций очень много (всего около 60-ти), мы решили сократить вывод и показать только несколько первых по алфавитному порядку. Это сделано с помощью команды

[1] Русскоязычная терминология в *Mathematica* не устоялась, поэтому авторы сами дают термины, эквивалентные принятым английским, и сами ими пользуются. А что делать?

Shallow, которая регулирует количество строк в клетке вывода (кстати, может быть использована и другая команда — Short):

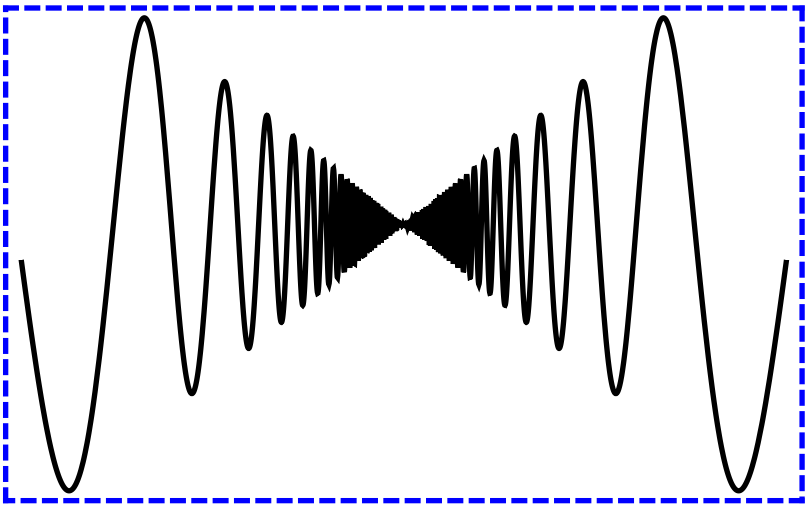
```
In[78]:= Options[Plot] // Shallow
Out[78]//Shallow=
    {AlignmentPoint → Center,
     AspectRatio → Power[≪2≫], Axes → True,
     AxesLabel → None, AxesOrigin → Automatic,
     AxesStyle → {}, Background → None,
     BaselinePosition → Automatic,
     BaseStyle → {≪3≫}, ClippingStyle → None, ≪47≫}
```

Все опции Plot так или иначе влияют на вид графика. Укажем, например, новый набор опций, одна из которых создает рамку (по умолчанию ее значение False), другая указывает стиль этой рамки и т. п.:

In[79]:= $\mathtt{Plot}\left[\xi\,\mathtt{Sin}\left[\frac{1}{\xi}\right],\ \left\{\xi,\ -\frac{\pi}{30},\ \frac{\pi}{30}\right\},\right.$

```
    Axes → None, FrameTicks → False, Frame → True,
    FrameStyle → Directive[Blue, Thick, Dashed],
    PlotStyle → {Thick, Black}]
```

Out[79]=



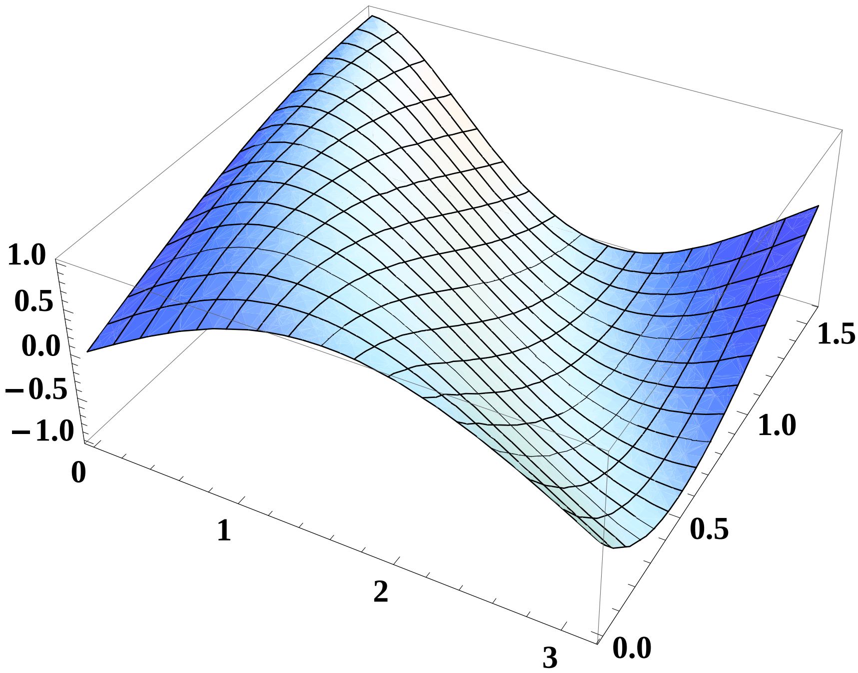
Сто́ит также обратить внимание на то, что именно график является ответом на заданный в клетке ввода вопрос. Интересно, что в предыдущих версиях *Mathematica*, до пятой включительно, рисунок являлся не основным, как можно было бы ожидать, а лишь *побочным результатом* (side effect) работы всех графических функций. Столь существенное изменение в работе системы связано с появлением в *Mathematica 6* одного из самых «могучих» ее инструментов на настоящий момент — функции `Manipulate`, которая позволяет, в частности, создавать разнообразные графические интерфейсы (graphical user interface — GUI). Огромная коллекция подобных изделий представлена на сайте *Mathematica* в разделе Demonstration Project. Полюбуйтесь на это, не пожалеете!

Теперь решим более сложную задачу — нарисуем поверхность, иначе говоря, построим график функции двух переменных. Это можно сделать с помощью `Plot3D`, имеющей три *позиционных* аргумента, смысл которых очевиден: сама функция и затем два списка, каждый из которых включает в себя имя переменной и границы области ее изменения. Выражение, являющееся первым позиционным аргументом в `Plot3D`, то есть изображаемая на графике функция, обязательно должно принимать *числовые* значения в прямоугольнике изменения переменных, заданном вторым и третьим аргументами — это условие общее для многих функций *Mathematica*. В противном случае появится сообщение об ошибке.

Опции служат мощным средством настройки рисунка по желанию пользователя. Конечно, у `Plot3D` есть свой, весьма обширный список опций, но сейчас мы посмотрим, как выглядят графики двумерных функций, когда все эти опции принимают значения, заданные по умолчанию:

In[80]:= **Plot3D[Sin[x + y + Log[x y + 1]], {x, 0, π}, {y, 0, $\frac{\pi}{2}$}]**

Out[80]=

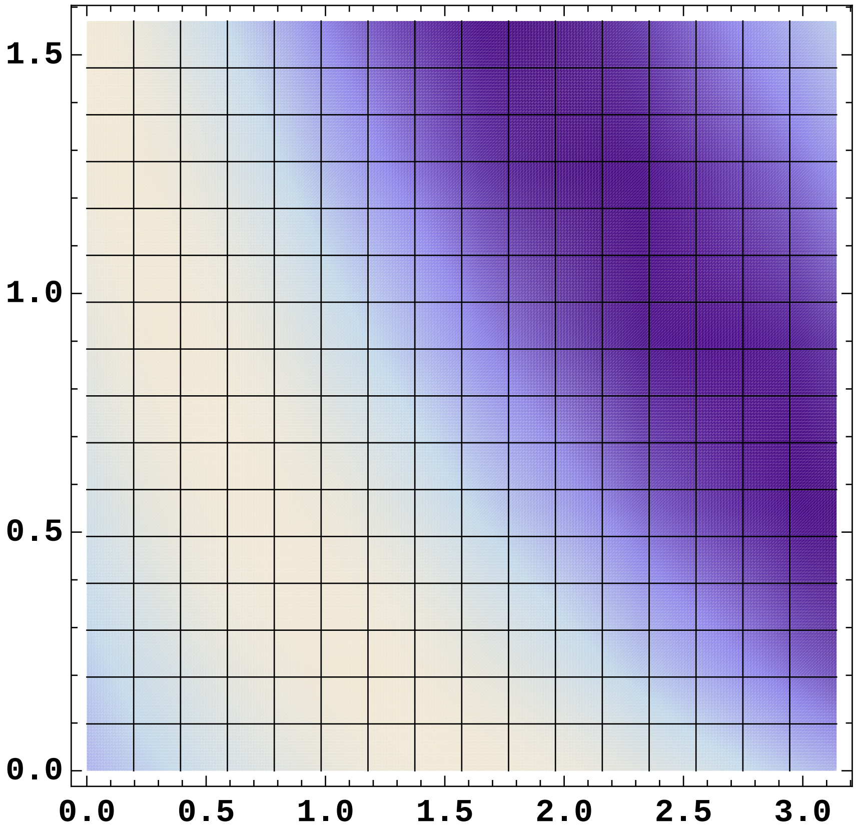


Обратите внимание, что участки поверхности окрашены в зависимости от их ориентации в пространстве. *Mathematica* «устанавливает» в заранее заданных точках несколько источников света (цветных «прожекторов»), которые и придают тот или иной оттенок различным областям графика функции. Цветовая гамма для изображаемой поверхности, «точка обзора» (опция ViewPoint), то есть точка, откуда мы смотрим на график, расстановка делений на осях и многое другое для данного графика соответствует значениям опций, принятым по умолчанию.

Вид сверху той же поверхности (функции двух переменных) создает графическая функция DensityPlot:

```
In[81]:= DensityPlot[Sin[x + y + Log[x y + 1]],
          {x, 0, π}, {y, 0, π / 2}, Mesh → True]
```

Out[81]=



На географических картах глубина морей характеризуется насыщенностью синего цвета. Так и здесь: чем участок поверхности ниже, тем в более темный цвет он окрашен.

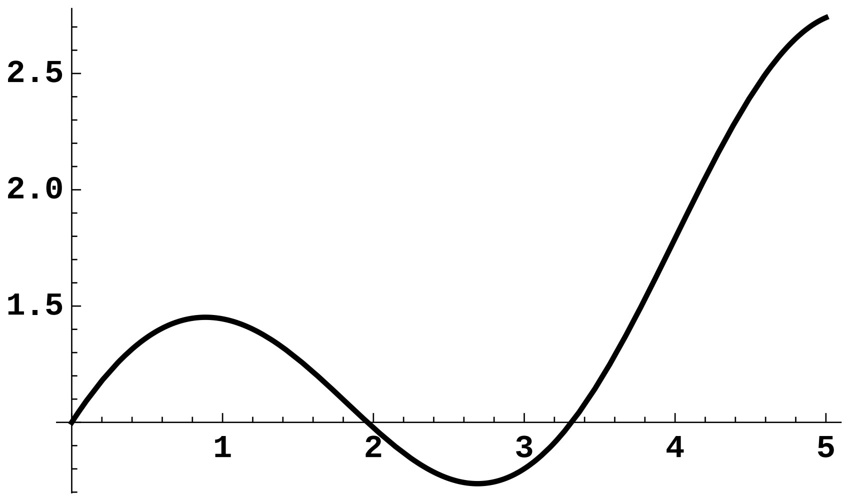
Вернемся к численному решению дифференциальных уравнений, которое реализовано в *Mathematica* функцией NDSolve. В отличие от DSolve, в первом аргументе следует *обязательно* задавать не только само дифференциальное уравнение, но и начальные условия, объединив их фигурными скобками в список. По сути, решается задача с любыми подходящими условиями, включая задачу Коши. Подходящими считаются условия, которые обеспечивают старт итераций численного метода. В «запасе» есть методы Рунге–Кутта, Адамса, Гира и многие-многие другие. Выбор метода осуществляется опцией `Method`. Вторым аргументом, как и в `DSolve`, является функция, которую мы ищем, а третий указывает переменную и допустимые границы ее изменения:

```
In[82]:= NDSolve[{y''[x] + √(x + 1) y[x] == x,
            y[0] == 1, y'[0] == 1}, y, {x, 0, 5}]

Out[82]= {{y → InterpolatingFunction[{{0., 5.}}, <>]}}
```

Mathematica получила ответ в виде правила подстановки (по виду это похоже на синтаксис опции: справа от стрелки указано найденное значение, которое сопоставлено переменной `y`) с так называемой интерполяционной функцией, определенной на интервале $[0;5]$. Присвоим переменной `sol` соответствующее значение и сразу же нарисуем график решения:

```
In[83]:= sol = First[y /. %]
         Plot[sol[x], {x, 0, 5},
          PlotStyle → {Black, Thick}]

Out[83]= InterpolatingFunction[{{0., 5.}}, <>]
```

Out[84]=



Теперь возьмем то же самое уравнение на том же самом интервале, но с тремя другими наборами условий. Обратите внимание, что в последнем случае сформулирована и численно решена краевая задача. Несложно определить соответствие решений и кривых, которые изображены на одном графике, — для этого они собраны в список, используемый как первый аргумент функции `Plot`:

```
In[85]:= NDSolve[{y''[x] + √(x + 1) y[x] == x,
            y[0] == 2, y'[0] == -1}, y, {x, 0, 5}];
```