Доля П.Г.
Харьковский Национальный Университет
механико – математический факультет
2015 г.

Mathematica для математиков.
Часть 1. Основы алгебраических вычислений

> Когда-то телескоп позволил людям увидеть дальше, а микроскоп ближе. *Mathematica* - инструмент сегодняшнего дня. Она позволяет увидеть дальше и ближе в математике и обнаружить идеи, которые являются новыми для нас, а иногда и для всего мира.
>
> Из книги Theodore W. Gray and Jerry Glynn. The Beginner's Guide to Mathematica. Version 2. (Addison-Wesley, 1992).

Система символьных вычислений *Mathematica* очень обширна и многогранна. Она содержит почти 4000 стандартных функций и необозримое число функций в пакетах расширения. В настоящем пособии мы даем описание только некоторых функций, которыми, как нам кажется, должен владеть каждый математик. Пособие предназначено в первую очередь для знакомства с математическими возможностями системы. Читайте текст и выполняйте примеры. Во многих случаях все пояснения дает само выполнение примера. Надеемся, что по окончании выполнения последнего примера вы начнете применять систему *Mathematica* для решения ваших задач. Уверяем, что в некоторых случаях вы будете удивляться получаемым решениям, но ваш математический кругозор после этого только расширится.

Примеры, приводимые в данном пособии, проверялись в версии пакета *Mathematica* 9.0. В более старых версиях часть примеров работать не будет. Особенности некоторых функций *Mathematica* 5 мы иногда описываем отдельно.

Большая часть, изложенного в пособии материала, доступна студентам младших курсов математических факультетов университетов, а также студентам технических вузов, прослушавшим курс высшей математики. Сложности могут возникнуть только при чтении параграфов, содержащих приложения системы *Mathematica* к решению прикладных задач, поскольку они предполагают некоторое знакомство с соответствующими областями знаний.

Весь материал многократно использовался в спецкурсе «Основы математических вычислений в системе *Mathematica*», читаемого автором студентам прикладного отделения механико-математического факультета Харьковского национального университета им. В.Н. Каразина, которые

специализируются по прикладной геометрии. Некоторые части этого пособия мы излагали в курсе информатики для студентов других отделений механико-математического факультета.

В настоящий момент (начало 2015 года) электронная версия пособия разделена на четыре части:

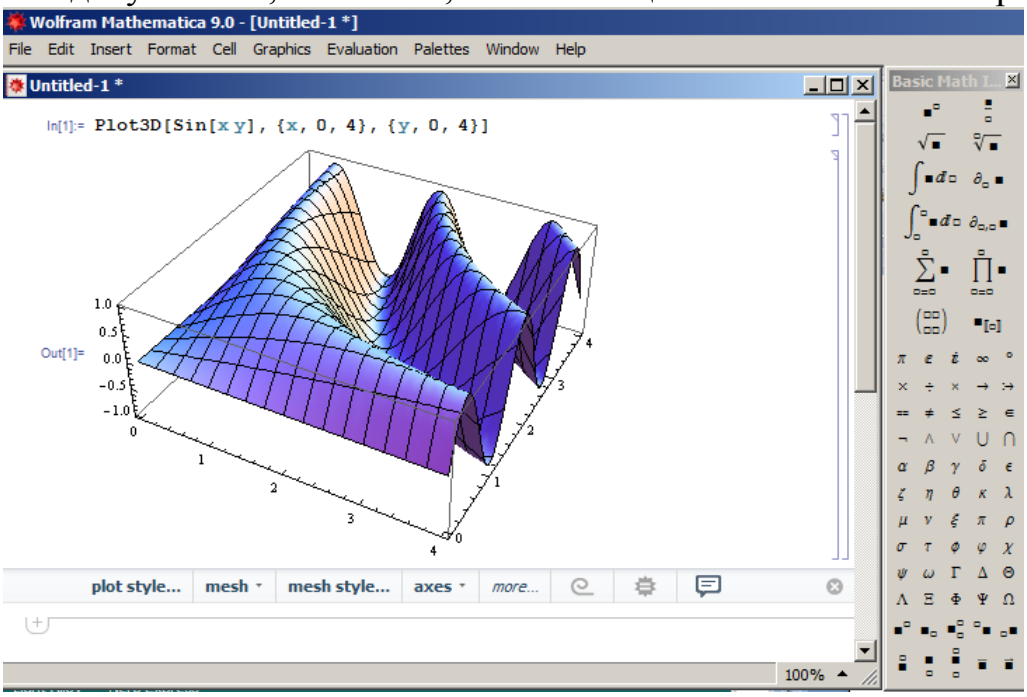
1. Основы алгебраических вычислений.
2. Графические возможности системы.
3. Реализация основных понятий математического анализа.
4. Решение дифференциальных уравнений.

Последние версии этих файлов вы можете найти на сайте geometry@karazin.ua в разделе автора «Документы».

Оглавление

Первые шаги

После запуска пакета Mathematica вы увидите стандартное меню программы, чистое окно документа и, возможно, панель специальных символов справа.



Внешний вид окна программы

Рис. 1.1

Наберите в окне документа: **2+2**. Теперь используйте комбинацию «горячих клавиш» **Shift-Enter** для того, чтобы Mathematica начала вычисления. Вы увидите:

In[1]:= **2 + 2**
Out[1]:=**4**

Справа от окна будут квадратные скобки, идентифицирующие разделы документа (секции) Отнеситесь к ним как к разметке различных листов бумаги, на которых вы делаете математические выкладки. Если вы хотите выполнить вычисление данной секции, курсор должен находиться в ее пределах. Символы стоящие слева In[1]:= и Out[1]:=, обозначают и нумеруют входные и выходные секции документа. Их наличие или отсутствие зависит от настроек программы.

В следующих разделах текст, набранный шрифтом **Arial-жирный** или стандартным формульным шрифтом системы *Mathematica* (стиль Input), вы можете вводить в окне документа и выполнять путем нажатия комбинации клавиш `Shift-Enter`.

Для математических операций в выражениях используются следующие знаки:

- ^ – обозначает возведение в степень; **2^3** это 2 в третьей степени.
- / – это деление; **34/89** это 34 делить на 89.
- * – это умножение; **34*89** это 34 умножить на 89.
- Пробел может служить для обозначения умножения: **34 89** это тоже, что и **34*89**. Однако пробелы часто ничего не обозначают; **Sin[x]** и **Sin [x]** это тоже что и **Sin[x]**.

- ( ) – (круглые скобки) служат для указания порядка вычислений: **(x+3)/x** это $x+3$ деленное на x. Не используйте квадратные или фигурные скобки для указания порядка вычислений.
- [ ] – (квадратные скобки) используются для указания аргументов функций; **Sin[x]** - это синус от x. Если написать Sin(x), то это не будет работать.
- { } – (фигурные скобки) обозначают список; **{1,2,3,4}** - это список из четырех чисел.
- % представляет результат последнего вычисления. Например

In[2]:= **3*7**
Out[2]:= **21**
In[3]:= **%^2**
Out[3]:= **441**

- **!** обозначает факториал; **5!** это 5 факториал.
- **=** обозначает присваивание; ***a*=5** это a присвоить 5. Значения переменных, присвоенные или вычисленные в одной секции, доступны в других секциях.
- **:=** отложенное присваивание.
- **==** проверка на равенство; ***a*==5** возвращает True, если a равно 5.
- != проверка на неравенство; ***a*!= 5** возвращает True, если a не равно 5. Следите за пробелами в этом выражении: ***a*!=5** может обозначать a факториал равно 5 или a не равно 5.
- <,<=,>,>= обозначают неравенства.

Набранная формула, как правило, не заканчивается никаким символом. В этом случае результат вычисления будет отображаться в окне документа. Если вы желаете выполнить вычисления, а результат не выводить в документ, то завершайте такие формулы точкой с запятой.

Основой пакета являются символьные вычисления. *Mathematica* умеет выполнять алгебраические преобразования, решать уравнения и системы уравнений, вычислять интегралы в аналитическом виде, решать дифференциальные уравнения, и умеет делать многие другие виды символьных вычислений, но и численные вычисления в ней выполняются великолепно.

Mathematica может выполнять арифметические операции не только как калькулятор. Она может выполнить алгебраическую операцию, где ответ не просто число.

In[4]:= $\mathbf{27x + 8x}$
Out[4]:= 35 x

Она может разложить выражение

In[5]:= $\mathbf{Expand[(a + b)\text{^}3]}$
Out[5]:= $343 + 147b + 21b^2 + b^3$

или

$\mathbf{Expand[(1 - x)(1 + x + x\text{^}2 + x\text{^}3 + x\text{^}4 + x\text{^}5)]}$
$1 - x^6$

Дальше мы не будем приводить идентификаторы In и Out входных и выходных секций документа. Кроме того, при переносе некоторых формул и результатов их вычисления из документа *Mathematica* в текстовый редактор, мы будем использовать команду копирования Copy As... – MathML (выделяем формулу,

щелкаем правой кнопкой мыши и в выпадающем меню выбираем Copy As... – MathML). Затем вставляем формулу в документ. В текстовом редакторе, в котором мы создаем наше пособие, такое копирование создает формулы в виде близком к привычной математической форме.

Мы можем проинтегрировать выражение по переменной *x*:

$$\int (a+x)^5 \, dx$$

Для набора формул в таком виде мы используем трафареты панели специальных символов Basic Math Input. Если панели нет на экране, то включить ее можно из меню: *Palettes – Other – Basic Math Input*. Результат последней команды будет следующим

$$\frac{1}{6}(a+x)^6$$

Символьные вычисления также имеют отношение к выражениям, в которые входят одни числа. Например, *Mathematica* может проводить вычисления с дробями и, приводя их к общему знаменателю, получать точный ответ.

$$\frac{1}{3}+\frac{2}{5}$$

$$\frac{11}{15}$$

Мы можем работать с числами большими настолько, что большинство калькуляторов их получить не могут, например, можно точно вычислить **200!**.

Основной принцип *Mathematica* – не делать никаких приближений кроме затребованных. Так выражение

$$\sqrt{12}$$

будет преобразовано к виду

$$2\sqrt{3}$$

Mathematica преобразует выражение, но оставляет в записи ответа квадратный корень, не выполняя никаких округлений. Если мы хотим получить численное приближение, то можем использовать функцию N.

$$N[\sqrt{12}]$$

3.4641

Функция *N*[…] может иметь два аргумента. Вторым необязательным аргументом можно определить точность результата (количество цифр)

$$N[\sqrt{12}, 40]$$

```
3.464101615137754587054892683011744733886
```

$$\mathbf{N}[\pi, 40]$$

```
3.141592653589793238462643383279502884197
```

Однако, если хотя бы одно из чисел задано с десятичной точкой, то и результат будет приближенным

$$\sqrt{12.}$$

```
3.4641
```

Если исходное число задано с точностью, превосходящей точность вычисления процессора (на персональных компьютерах с 32 – х битовым процессором это примерно 15 значащих цифр), то и результат будет иметь точность исходных данных. Сравните

$\sqrt{12.000000}$

```
3.4641
```

$\sqrt{12.0000000000000000}$

```
3.4641016151377546
```

Все встроенные в Mathematica имена начинаются с заглавной буквы (**Sin, Table, Factor** и т.д.), некоторые имеют несколько заглавных букв **ContourPlot**. Вы должны писать эти имена так, как показано или код не будет работать.

Аргументы функций заключают в квадратные скобки, а не круглые (**Sin[x], Factor[x^2-9]** и т.д.). Если вы хотите получить правильный ответ, не используйте **Sin(x)** или **sin(x)** или **sin[x]**, вы должны использовать только **Sin[x].**

Многие специальные символы, для которых нет соответствующих кнопок на клавиатуре, можно ввести, используя панель специальных символов, которая обычно расположена справа от окна документа (см. рисунок 1.1). Она позволяет вводить математические формулы в виде близком к естественной математической записи. Имеются и другие панели. Вывести панель на экран можно из меню *Palettes,* в котором следует выбрать имя – название одной из панелей. Панель специальных символов, показанная на рис. 1.1, имеет имя *BasicMathInput* и вызывается из меню *Palettes – Other – Basic Math Input.*

На панели спецсимволов (имя панели – *BasicMathInput*) имеются знаки интеграла $\int$, ряда $\sum$, корня $\sqrt{\ }$ и многие другие. При выборе мышью одного из них в окне документа появляется трафарет ввода, который надо заполнить обычными символами

$\int \mathbf{Sin}[x]\, dx$

$-\mathrm{Cos}[x]$

Однако возможен ввод математических знаков с помощью специальных имен функций. Последний пример можно выполнить так:

$\mathbf{Integrate}[\mathbf{Sin}[x], x]$

Ядро пакета *Mathematica* понимает только такие имена функций, а оболочка переводит специальные обозначения, например $\int \mathrm{Sin}[x]\mathrm{dx}$, в понятный ядру текст Integrate[Sin[x],x].

Вы можете набирать математические формулы либо в привычном математическом виде, используя спецсимволы, либо в текстовом формате, как это было показано в предыдущем примере для интеграла. Для набора большинства специальных символов есть особые комбинации клавиш.

Mathematica умеет работать с комплексными числами. В следующем примере символ мнимой единицы возьмите с панели специальных символов

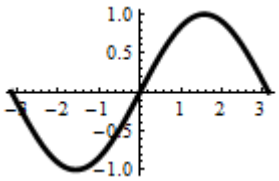
$(\mathbf{2} + i)(\mathbf{10} - i)$

$21 + 8i$

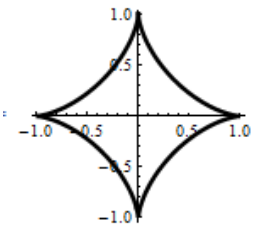
Для запоминания результатов вычислений удобно использовать переменные. При определении переменной (константы) используется знак равенства. Слева от знака равенства стоит имя переменной, а справа – ее значение или выражение для вычисления.

$$a = 5^3 + 4\frac{12 + 2^5}{2^4}$$
136
Такую константу можно использовать в различных выражениях:
$$a + 7$$
143
$$(a - 130)^3$$
216
Если длина формулы больше ширины окна документа, то система автоматически выполнит перенос части формулы на следующую строку. Никаких специальных знаков продолжения строки не используется.

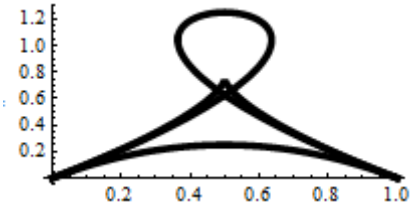
Можно создавать свои функции. При определении функции необходимо использовать подчеркнутые символы после каждого имени аргумента в левой части и знак отложенного присваивания := (или обычного присваивания =) в середине. Справа от этого знака должно стоять выражение для вычисления значения функции.
$$f[x\_] := x^2$$
Эту функцию можно вызывать с разными типами аргументов, например, с числовым аргументом
$$f[5]$$
25
Мы можем использовать ее с символьным выражением
$$f[c + b]$$
$$(b + c)^2$$
Можно построить график этой функции
$$\mathbf{Plot}[f[x], \{x, -2, 2\}]$$
Можно взять производную этой функции
$$f'[u]$$
$$2u$$
Графические возможности пакета весьма обширны. Для первого знакомства мы приведем только несколько примеров. Наберите точно такие команды, и вы получите аналогичные рисунки
$$\mathbf{Plot}[\mathbf{Sin}[x], \{x, -\pi, \pi\}]$$



$$\mathbf{ParametricPlot}[\{\mathbf{Sin}[t]^3, \mathbf{Cos}[t]^3\}, \{t, 0, 2\mathbf{Pi}\}]$$



ParametricPlot[{
$\{t, t-t^2\}, \{3t-6t^2+4t^3, 3t-3t^2\},$
$\{5t-12t^2+8t^3, 5t-5t^2\}\}, \{t, 0, 1\},$
AspectRatio → **0.5**, **PlotStyle** → {{**Black**, **Thickness**[**0.02**]}}]



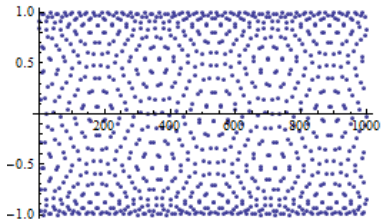
В последнем примере были построены три кривые, уравнения которых заданы в параметрическом виде (стрелочка набирается как знак минус и знак больше). В нашем тексте здесь и далее длинные команды занимают две или более строк. В *Mathematica* набирайте их одной строкой или выполняйте перенос на следующую строку нажатием клавиши Enter.

Для генерирования случайных чисел используется функция `Random[]` (по умолчанию в диапазоне от 0 до 1)

Random[ ]

Вот как можно построить график по некоторому набору точек

ListPlot[**Table**[**Sin**[i], {i, **1**, **1000**}]]

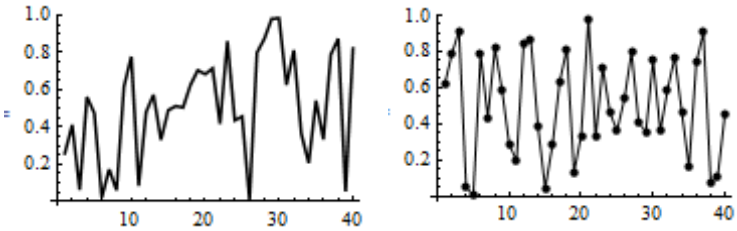


Если точки нужно соединять отрезками, то следует использовать опцию (необязательный аргумент) `Joined→True` (следующий рисунок слева).

ListPlot[**Table**[**Random**[], {**40**}], **Joined** → **True**,
PlotStyle → {{**Black**, **Thickness**[**0.01**]}}]

Здесь опция `Joined→True` повлияла на способ отображения графика – вместо точек была построена ломаная. Если вы хотите отобразить точки и ломаную, то следует добавить еще две опции Mesh → All и MeshStyle → PointSize[размер]. Например, следующий рисунок справа построен командой)

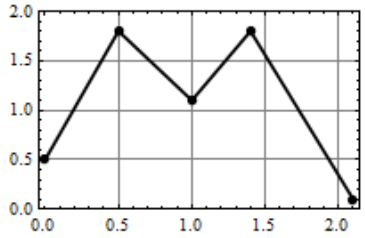
ListPlot[**Table**[**Random**[], {**40**}], **Joined** → **True**,
PlotStyle → {{**Black**, **Thickness**[**0.01**]}}, **Mesh** → **All**,
MeshStyle → **PointSize**[**0.03**]]



Заметим, что в Mathematica 5 опция, управляющая соединением точек, имела вид `PlotJoined→True`.

Вот еще один пример.

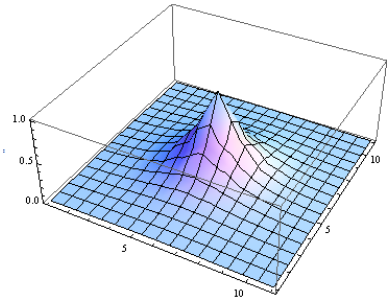
```
lst = {{0, 0.5}, {0.5, 1.8}, {1., 1.1}, {1.4, 1.8}, {2.1, 0.1}};
ListPlot[lst, Joined → True, Frame → True, PlotRange → {0., 2.},
    GridLines → Automatic, Mesh → All, MeshStyle → PointSize[0.03],
    PlotStyle → {{Black, Thickness[0.01]}}]
```



Здесь функции `ListPlot` в качестве аргумента передан список `lst`. Остальные аргументы (опции) не являются обязательными и передаются функции в виде: имя → значение. О них мы поговорим позже.

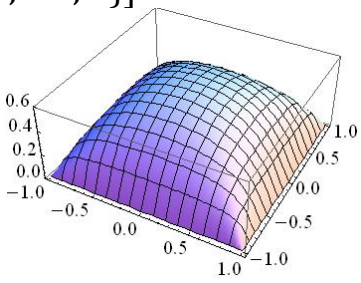
Вот пример табулирования точек поверхности

```
a1 = Table[1/(1 + x^2 + y^2), {x, -5, 5, 1}, {y, -5, 5, 1}];
ListPlot3D[a1, PlotRange → All]
```



Можно создавать функции двух переменных и строить поверхности по их уравнению

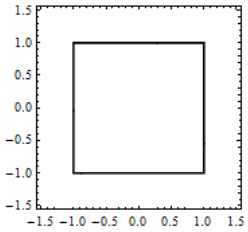
```
W[x_, y_] := 2 - x^2 - y^2 - Sqrt[(1 - x^2)^2 + (1 - y^2)^2]
Plot3D[W[x, y], {x, -1, 1}, {y, -1, 1}]
```



Здесь первым аргументом функции `Plot3D` является выражение, график которого следует построить, вторым и третьим – диапазоны изменения независимых переменных.

В следующем примере строится квадрат по его неявному уравнению W[x,y]=0, где выражение для W используется из предыдущего примера (в *Mathematica* это будет предыдущая секция документа). Обратите внимание на двойной знак равенства.

```
ContourPlot[W[x, y] == 0, {x, -1.5, 1.5}, {y, -1.5, 1.5}, PlotPoints → 100,
            ContourStyle → {{Black, Thickness[0.01]}}]
```



(В *Mathematica* 5 вместо функции `ContourPlot` использовалась функция `ImplicitPlot`, которая находится в пакете расширения `<<Graphics`ImplicitPlot`.` Также вместо опции `ContourStyle` использовалась опция `PlotStyle` с теми же значениями).

Многие функции системы *Mathematica* находятся в пакетах расширения – специальных программах, подключаемых во время работы. Чтобы выполнить такие функции надо загрузить соответствующий пакет командой **<<** (подряд два знака меньше) с указанием имени пакета. Например, `<<ComputationalGeometry`.` Здесь используются обратная одинарная кавычка. Ее кнопка находится на клавиатуре под клавишей ESC. Для загрузки пакета можно также использовать функцию Needs. Например, Needs["ComputationalGeometry`"]. В этой записи используются двойные и обратная одинарная кавычки.

В следующем примере с помощью функции `PlanarGraphPlot[…]` строится триангуляция Делоне. Код функции находится в пакете расширения `ComputationalGeometry`` (обратная косая кавычка в конце имени обязательна). Координаты точек задаются в списке `data`.

```
<< ComputationalGeometry`
data = {{4.4, 14}, {6.7, 15.25}, {6.9, 12.8}, {2.1, 11.1}, {9.5, 14.9},
        {13.2, 11.9}, {10.3, 12.3}, {6.8, 9.5}, {3.3, 7.7}, {0.6, 5.1},
        {5.3, 2.4}, {8.45, 4.7}, {11.5, 9.6}, {13.8, 7.3}, {12.9, 3.1}, {11, 1.1}};
PlanarGraphPlot[data, LabelPoints → False]
```

Если надо использовать функции пакета расширения еще раз, то повторной загрузки пакета не требуется. Загруженные функции будут доступны в течении всего сеанса работы с текущим документом *Mathematica*.

Система символьных вычислений *Mathematica* как программа состоит из четырех основных частей – оболочки, ядра (*Kernel*), набора пакетов расширения и справочной системы. Оболочка – это интерфейс программы, которую мы видим на экране и которую используем для набора формул. Ядро – это динамически загружаемая библиотека функций, которою можно

использовать даже из других программ, и функции которой выполняют все вычисления. Пакеты расширения содержат большую часть функциональных возможностей системы. Их может создавать любой пользователь.

Справочная система вызывается из меню Help – Documentation Center или клавишей F1. Если в документе выделить название какой-нибудь функции и нажать F1, то откроется окно со справкой по этой функции. Примеры можно выполнять в окне справки так же, как и в окне основного документа, нажатием комбинации клавиш Shift-Enter.

Работа с «Математикой» оформляется в виде отдельного файла, который имеет расширение имени *.nb. Файл разбит на ячейки, ячейки имеют стиль, который определяет, что можно делать с содержимым ячейки. Например, ячейки, содержащие вычисляемые выражения имеют стиль «Input», ячейки, содержащие результат, имеют стиль «Output». Бывают ячейки со стилем «Text» или с заголовочными стилями. Для выполнения вычислений в ячейке используется комбинация клавиш `Shift-Enter` или можно использовать клавишу `Enter` на цифровой области клавиатуры. После выполнения вычислений в секции после нее появляется «Панель предложения дальнейших действий», на которой мы можем выбрать что мы хотим сделать далее.

Двойной щелчок по графику и затем `Ctrl – d` откроет палитру рисования по графику.

Знакомство с другими возможностями системы вы найдете в следующих разделах данного пособия.

1. Основы алгебраических вычислений

1.1 Основные типы данных и переменные

В пакете `Mathematica` вы можете использовать любые знакомые вам из алгебры типы данных: целые числа, рациональные, иррациональные, комплексные. С основными из них вы познакомились в предыдущем разделе. Осталось показать, как работать с комплексными числами и переменными.

Для ввода комплексных чисел можно использовать стандартную нотацию, используя для мнимой единицы символ I (прописная буква I) или специальный символ $\boldsymbol{ii}$, который можно ввести последовательной комбинацией клавиш Esc – **i** – **i** - Ecs (два раза символ i), а также с панели спецсимволов. Знаки операций используются те же, что и для вещественных чисел.

$$\boldsymbol{a = 2 + I; b = 3 - I; c = a * b}$$

$$\boldsymbol{d = a/b}$$

$$7 + ii$$

$$\frac{1}{2} + \frac{ii}{2}$$

Напомним, что точка с запятой в конце выражения отменяет вывод результатов расчета в окно документа. Поэтому значения переменных a и b при выводе не повторяются.

Функции `Re[z]` и `Im[z]` вычисляют вещественную и мнимую часть комплексного числа `z`.

$$a = 2 + I; b = 3 - I; c = a * b$$

$$z = a + b * c - c/a$$

$21 - 2i$

$\mathbf{Re}[z]$

$\mathbf{Im}[z]$

21

-2

Функция `Abs[z]` и `Arg[z]` вычисляют модуль и аргумент комплексного числа `z`.

$\mathbf{Abs}[z]$

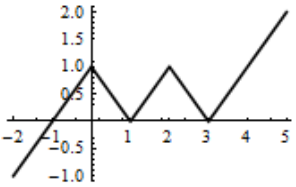
$\mathbf{Arg}[z]$

$\sqrt{445}$

$-\text{ArcTan}[\frac{2}{21}]$

Естественно, что функция `Abs[z]` используется и для вычисления модулей вещественных чисел.

$$\mathbf{Plot}[x - 1 - \mathbf{Abs}[x] + \mathbf{Abs}[x - 1] - \mathbf{Abs}[x - 2] + \mathbf{Abs}[x - 3], \{x, -2, 5\}]$$



Переменная может хранить любой тип данных – число, символ, список, матрицу, функцию, текст, графический фрагмент, звук и т.д. Для работы с ней можно использовать любые допустимые для такого типа данных функции. Например, любое имя, не имеющее никакого значения, интерпретируется системой как идентификатор (заменитель) числа и для работы с ним используются алгебраические функции. Любой набор имен можно перемежать знаками арифметических операций и выполнять над ним стандартные алгебраические преобразования.

Особое значение имеет операция присваивания. В системе имеется две операции: « = » – непосредственное присваивание и « := » – отложенное присваивание. В выражении `name=expr` происходит вычисление правой части выражения `expr` и сразу же результат присваивается переменной `name`. Вычисление правой части `expr` выражения отложенного присваивание `name:=expr` выполняется в момент, когда потребовалось значение левой части, т.е. *name* потребовалось для вывода или вычисления Для пояснения разницы рассмотрим следующую последовательность команд, содержащую непосредственное присваивание.

$$a = 5; x = a^2; a = 6; x$$

25

Здесь мы присвоили значение 5 переменной *a*, затем переменной *x* присвоили значение a^2. В результате непосредственного присваивания произошло

вычисление правой части выражения $x = a^2$ и левой части, т.е. переменной *x*, было присвоено значение 25. Последующее изменение значения переменной *a* не повлияло на значение *x*. Выполним туже последовательность команд, но для переменной *x* выполним отложенное присваивание (предварительно удалив прежнее значение *x* из рабочего пространства системы).

$\boldsymbol{x =.;\ a = 5;\ x := a^2; a = 6; x}$

36

Здесь присваивание $x = a^2$ выполнялось в момент вывода значения переменной *x*. Но в этот момент переменная *a* равнялась 6 и, поэтому, мы получили $x = 6^2$, т.е. 36.

Еще раз. Отложенное присваивание *x*:=f[*a*] означает, что значение f[*a*] не будет присвоено переменной *x* до тех пор, пока *x* не потребуется. Это означает, что в момент использования переменной *x* будет использовано последнее значение переменной *a*.

Кроме присваивания вам, наоборот, иногда потребуется удалять имена переменных. Для этого используется функция `Remove[x]`. Она удаляет из рабочего пространства системы символ *x* так, что его имя больше не распознается системой, в частности, не распознается тип переменной. Функция `Clear[x]` очищает значение и удаляет определение символа `x`. Удалить значение одной переменной можно командой `x=.` (`x` присвоить току), а функциям `Remove` и `Clear` можно передавать имена нескольких переменных, перечисляемых через запятую.

Выполните следующие цепочки команд и проанализируйте результат.

$\mathbf{Clear}[f, x]; x = 3; f = x; x = 1; \{x, f\}$

```
{1,3}
```

$\mathbf{Clear}[f, x]; f = x; x = 3; x = 1; \{x, f\}$

```
{1,1}
```

$\mathbf{Clear}[f, x]; x = 3; f = x; x = 1; \mathbf{Clear}[x]; \{x, f\}$

```
{x,3}
```

$\mathbf{Clear}[f, x]; f = x; x = 3; x = 1; \mathbf{Clear}[x]; \{x, f\}$

```
{x,x}
```

$\mathbf{Clear}[f, x]; f = x; x = 3; x = 1; \mathbf{Remove}[x]; \{x, f\}$

```
{Removed[x],Removed[x]}
```

Любой набор чисел или имен, заключенный в фигурные скобки, интерпретируется как список или даже вектор и над ним можно выполнять преобразования естественные для такого типа данных. В следующих разделах пособия вы познакомитесь с этим и другими (нечисловыми) типами данных, и функциями для работы с ними.

1.2 Работа со списками, векторами и матрицами

В пакете *Mathematica* нет векторов и матриц, но есть списки, которые их заменяют. Любой набор элементов заключенный в фигурные скобки является списком. Список списков заменяет матрицу. Возможно и более глубокое вложение списков. Здесь мы рассматриваем основные операции, которые можно выполнять со списками.

Список можно складывать с числом. В этом случае число прибавляется к каждому элементу списка.

```
5+{10,20,30,40}
```

$\{15,25,35,45\}$

Список можно умножить на число.

```
5 {10,20,30,40}
```

$\{50,100,150,200\}$

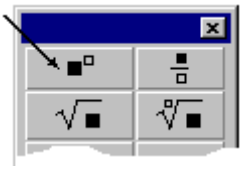
В этом случае каждый элемент списка умножается на соответствующее число.

Можно возвести в квадрат каждый элемент списка.

```
{10,20,30,40}^2
```

$\{100,400,900,1600\}$

Отметим, что возведение в степень можно выполнять с помощью операции ^ (шляпка) или с использованием специального трафарета панели специальных символов, который также можно ввести комбинацией клавиш Ctrl - ^.



Там, где может стоять число или переменная может стоять список.

$$\mathbf{Sin}\left[\left\{\frac{\boldsymbol{\pi}}{\mathbf{6}},\frac{\boldsymbol{\pi}}{\mathbf{4}},\frac{\boldsymbol{\pi}}{\mathbf{3}},\frac{\boldsymbol{\pi}}{\mathbf{2}}\right\}\right]$$

$$\left\{\frac{1}{2},\frac{1}{\sqrt{2}},\frac{\sqrt{3}}{2},1\right\}$$

Можно объединить вместе два списка.

```
Join[{10,20,30,40},{100,200,300,400}]
```

$\{10,20,30,40,100,200,300,400\}$

Мы можем получить элемент списка (двойные квадратные скобки обозначают извлечение элемента).

$$\{\mathbf{10},\mathbf{20},\mathbf{30},\mathbf{40}\}[[\mathbf{2}]]$$

20

или

```
lst={10,20,30,40};
lst[[3]]
```

30

Можно выделить часть списка (промежуток)

$$\boldsymbol{L}=\{\mathbf{3},\mathbf{5},\mathbf{1},\mathbf{2},-\mathbf{2},\mathbf{0},\mathbf{7},\mathbf{8}\}$$

$$\boldsymbol{L}[[\mathbf{3};;\mathbf{6}]]$$

$\{1,2,-2,0\}$

Можно присваивать один список другому

$\{a, b, c\} = \{x^2, x\text{^}2 - 3x + 2, \frac{x+1}{x-1}\}$

a

b

c

x^2

$2 - 3x + x^2$

$$\frac{1 + x}{-1 + x}$$

При выполнении такого присваивания происходит вычисление выражений в правой части и затем присваивание результата элементам левого списка.

Можно сложить поэлементно два списка. В этом случае списки должны иметь одинаковое количество элементов.

{10,20,30,40}+{100,200,300,400}

```
{110,220,330,440}
```

Можно выполнить следующее сложение списков

{1,3,5}+{{2,1,3},{5,6,7},{3,2,3}}

```
{{3,2,4},{8,9,10},{8,7,8}}
```

Здесь каждый элемент первого списка складывается с соответствующим внутренним списком второго слагаемого.

Можно умножить поэлементно два списка. Это умножение не является скалярным произведением.

{10,20,30,40} {100,200,300,400}

```
{1000,4000,9000,16000}
```

Список может быть реорганизован:

Reverse[{10,20,30,40}] (обращение списка)

```
{40,30,20,10}
```

Sort[{4,2,3,1}] (сортировка списка по возрастанию)

```
{1,2,3,4}
```

Reverse[Sort[{4,2,3,1}]] (сортировка списка по убыванию)

```
{4,3,2,1}
```

Элементы списка не обязательно должны быть числами.

{x, x^2, x^3, x^4, Sin[x]}

Если элементами списка являются числа, то их можно представить графически.

ListPlot[{**4, 2, 3, 1, 5, 2, 5, 8, 4, 2, 4, 4, 2**}, **PlotStyle → PointSize**[**0. 02**]]

При построении графика этого списка абсциссами точек (узлов) являются числа {1,2,3,…} – номера элементов списка, а ординатами – значения элементов. Отметим, что точки, попавшие на оси координат незаметны из-за их размеров. Размер точек на графике управляется опцией PlotStyle → PointSize[0.02].

Можно организовать список списков. В общем случае можно представить список пар чисел:

{{**1, 5**}, {**4, 2**}, {**1, 3**}, {**3, 3**}, {**7, 1**}, {**5, 5**}, {**3, 3**}, {**2, 5**}}

Такой список может функцией `ListPlot` будет интерпретироваться как список *x, y* координат точек.

ListPlot[%]

Напомним, что символ % представляет объект последнего вычисления системы, не предыдущий в документе, а последний вычисленный.

Существует много команд, связанных с обработкой списков. Вот некоторые из них: `First, Last, Rest, Part, Take, Drop, Append, Insert, Delete`.

$\boldsymbol{L = \{3, 5, 8, 1, 9, 0, -2\}}$

Part[$\boldsymbol{L}$, **5**] – выбирает 5-й элемент списка (=L[[5]]);

First[$\boldsymbol{L}$] – выбирает первый элемент списка L;

Last[$\boldsymbol{L}$] – выбирает последний элемент списка L;

$\boldsymbol{L}$[[{**1**, **3**, **5**}]] – возвращает список, составленный из элементов L с указанными номерами.

Смысл других функций понятен из их названия. Заметим, что эти функции не реорганизуют список L, а возвращают новый список. Например, функция `Rest[L]`, создает новый список, являющийся копией списка L без его первого элемента.

Получить краткую справку по любой функции можно, например, так:

```
?Delete
```

В результате в окне документа появится краткая информация об этой функции. Справку по функции можно открыть в отдельном окне справочной системы. Для этого надо внутри набранного в окне документа имени интересующей вас функции поместить курсор и нажать клавишу F1.

Списки можно создавать. Вот пример создания списка целых чисел.

```
Range[3,16,2]
{3,5,7,9,11,13,15}
```

Аргументами функции `Range` являются начальное значение списка, максимально допустимое значение и шаг.

Чтобы просуммировать элементы списка, можно использовать функцию `Sum`. Ее первым аргументом является формула вычисления общего члена, а вторым – список, указывающий имя изменяемой переменной и ее максимальное значение.

```
L:={1,5,3,4}
Sum[ L[[ i ]], {i,4}]
13
```

Вот еще примеры работы со списком. Введите в одной секции документа следующий код (без наших пояснений)

```
L1={1,2,4,3,8,6,7};
L2={{1,2,3},{4,5,6},{7,8,10}};
```

L1[[{1,3,5}]] - из списка `L1` выбрать 1-й, 3-й и 5-й элементы;

L2[[2,3]] - из `L2` выбрать 2-й внутренний список и из него взять 3-й элемент;

Det[L2] - вычислить детерминант.

Вы увидите следующий результат:

```
{1,4,8}
 6
-3
```
Функция `Length` возвращает длину списка (число элементов верхнего уровня). Функция `Dimensions` возвращает список - количество элементов по каждому из индексов входного списка. Функция `TensorRank` возвращает количество индексов (размерность) списка.

Length[L1]
Dimensions[L1]
Dimensions[L2]
TensorRank[L2]

Поучаем следующий результат.

```
7
{7}
{3,3}
2
```

Имеются различные функции для автоматического создания списков. В следующем примере создается таблица квадратов целых чисел:

Table[n^2, {n, 1, 10}]

```
{1,4,9,16,25,36,49,64,81,100}
```

Первый аргумент - выражение для формирования элементов списка. Второй аргумент - «итерационная спецификация». Здесь она означает, что **n** изменяется от 1 до 10 с шагом 1. Мы можем изменить размер шага, добавив четвертый параметр в список:

Table[n^2, {n, 1, 10, ½}]

Двойной список, можно интерпретировать как прямоугольную матрицу и выполнять с ним естественные для таких объектов операции (транспонирование и т.д.).

e1=Table[{t, t^2,}, {t, 0, 5, 1}]

```
{{0,0}, {1,1}, {2,4}, {3,9}, {4,16}, {5,25}}
```

Transpose[e1]

```
{{0,1,2,3,4,5}, {0,1,4,9,16,25}}
```

Двойной список может быть реорганизован в одинарный.

Flatten[e1]

```
{0,0,1,1,2,4,3,9,4,16,5,25}
```

У функции `Flatten` есть вариант вызова Flatten$[L, n]$, где n представляет уровень реорганизации

$\boldsymbol{L = \{\{\{1,2,3\},\{4,5,6\},\{7,8,9\}\},\{\{1,1,0\},\{0,2,2\},\{0,0,0\}\}\}}$

$\mathbf{Flatten}[\boldsymbol{L}, \mathbf{0}]$

$\mathbf{Flatten}[\boldsymbol{L}, \mathbf{1}]$

$\mathbf{Flatten}[\boldsymbol{L}, \mathbf{2}]$

$\mathbf{Flatten}[\boldsymbol{L}]$

$\{\{\{1,2,3\},\{4,5,6\},\{7,8,9\}\},\{\{1,1,0\},\{0,2,2\},\{0,0,0\}\}\}$

$\{\{1,2,3\},\{4,5,6\},\{7,8,9\},\{1,1,0\},\{0,2,2\},\{0,0,0\}\}$

$\{1,2,3,4,5,6,7,8,9,1,1,0,0,2,2,0,0,0\}$

$\{1,2,3,4,5,6,7,8,9,1,1,0,0,2,2,0,0,0\}$

Как видим, вызов `Flatten` без второго аргумента эквивалентен вызову с максимальным значением уровня реорганизации.

Можно создавать список символьных выражений.

Table[Expand[$(1+x)^n$], {n, 1, 3}]

$\{1+x, 1+2x+x^2, 1+3x+3x^2+x^3\}$

Мы можем создать список списков.

Table[{n, n^2, n^3, n^4}, {n, 1, 3}]

```
{{1,1,1,1},{2,4,8,16},{3,9,27,81}}
```

или

Table[n^a, {n, 1, 3}, {a, 1, 4}]

```
{{1,1,1,1},{2,4,8,16},{3,9,27,81}}
```

Первый аргумент - выражение, которое определяет каждый элемент списка. Второй аргумент - «медленный» итератор (изменяется при переходе от одного внутреннего списка к другому). Третий аргумент - «быстрый» итератор (изменяется при переходе от элемента к элементу внутри каждого внутреннего списка).

В пакете *Mathematica* имеется возможность префиксного (перед аргументом) и постфиксного (после аргумента) использования имен функций одной переменной. Если имя функции используется перед аргументом, то аргумент надо заключать в квадратные скобки, например, `Sin[Pi/4]`. Если имя функции стоит после аргумента, то аргумент отделяется от него двойной косой чертой, например, `Pi/4 // Sin`. Оба способа вернут одинаковый результат $\frac{1}{\sqrt{2}}$.

Функция `TableForm` показывает списки списков в двумерном виде. Ее удобно использовать в постфиксной нотации. Сравните

`TableForm[{{a,b},{c,d}}]` или `{{a,b},{c,d}} // TableForm`

```
a  b
c  d
```

Здесь запись `{{a,b},{c,d}} // TableForm` означает применение функции `TableForm` к списку `{{a,b},{c,d}}`.

Вот пример создания таблицы чисел 3×4.

TableForm[Table[n^a, {n, 1, 3}, {a, 1, 4}]]

```
1  1   1    1
2  4   8   16
3  9  27   81
```

Вектор в *Mathematica* - это список {x, y}. *Mathematica* не различает вектора-строки и вектора-столбцы. Список может быть и тем, и другим, в зависимости от контекста. Матрица в *Mathematica* - это список списков

$\{\{a, b\}, \{c, d\}\}$

Чтобы увидеть более традиционную запись, можно добавить вызов функции `MatrixForm`.

MatrixForm[{{a,b},{c,d}}] или **{{a,b},{c,d}}// MatrixForm**

$$\begin{pmatrix} a & b \\ c & d \end{pmatrix}$$

При вводе матриц добавление строк выполняется комбинацией клавиш Ctrl - Enter, а добавление столбцов Ctrl -,(запятая).

Функция Dot может выполнить матричное умножение списков (матриц), умножение матрицы на вектор или скалярное умножение векторов.

Dot[{{1,2},{3,4}},{{1,4},{2,5}}]

```
{{5,14},{11,32}}
```

Синонимом функции Dot является оператор '**.**' (точка). С его помощью мы можем определить произведение векторов и/или матриц.

{{a,b},{c,d}}.{x,y}

```
{ax+by,cx+dy}
```

или

{x,y}.{{a,b},{c,d}}

```
{ax+cy,bx+dy}
```

В первом случае {x,y} выступает, как вектор-столбец, во втором - как вектор-строка. В следующем выражении присутствуют оба типа.

$\{\boldsymbol{x},\boldsymbol{y}\}.\{\boldsymbol{x},\boldsymbol{y}\}$

x^2+y^2

Можно использовать привычное обозначение векторов и матриц, но результат все равно будет списком списков. Если вы желаете его увидеть в виде вектора или матрицы используйте функцию MatrixForm.

$(\boldsymbol{x}\quad \boldsymbol{y}).\begin{pmatrix}\boldsymbol{x}\\ \boldsymbol{y}\end{pmatrix}$

$\{\{x^2+y^2\}\}$

Для выполнения предыдущей команды наберите круглые скобки (обе, открывающую и закрывающую), внутри скобок наберите x, затем нажмите комбинацию клавиш Ctrl-запятая, затем введите y. Переместите курсов вправо за пределы скобок, введите точку, наберите снова открывающую и закрывающую скобки. Внутри этих скобок введите x, затем наберите комбинацию клавиш Ctrl-Enter, затем введите y. Аналогично выполняется набор следующей команды

$\begin{pmatrix}\boldsymbol{a} & \boldsymbol{b}\\ \boldsymbol{c} & \boldsymbol{d}\end{pmatrix}.\begin{pmatrix}\boldsymbol{e}\\ \boldsymbol{f}\end{pmatrix}$ **//MatrixForm**

$\begin{pmatrix}a\,e+b\,f\\ c\,e+d\,f\end{pmatrix}$

Рассмотрим некоторые встроенные функции для манипуляции векторами и матрицами. Все эти функции работают и с числовыми, и с символьными матрицами. Мы будем демонстрировать их на простых символьных матрицах, так что вы увидите результат.

Вычисление детерминанта

$\mathbf{Det}[\{\{\boldsymbol{a},\boldsymbol{b}\},\{\boldsymbol{c},\boldsymbol{d}\}\}]$

$-b\,c+a\,d$

Команда

$\mathbf{Det}[\mathbf{Table}[\boldsymbol{a}[\boldsymbol{i},\boldsymbol{j}],\{\boldsymbol{i},\mathbf{1},\mathbf{3}\},\{\boldsymbol{j},\mathbf{1},\mathbf{3}\}]]$

даст формулу вычисления детерминанта матрицы 3 x 3

$$-a[1,3]a[2,2]a[3,1]+a[1,2]a[2,3]a[3,1]+a[1,3]a[2,1]a[3,2]-$$
$$a[1,1]a[2,3]a[3,2]-a[1,2]a[2,1]a[3,3]+a[1,1]a[2,2]a[3,3]$$

Можно вычислить след матрицы

$\boldsymbol{M}=\mathbf{Table}[\boldsymbol{a}[\boldsymbol{i},\boldsymbol{j}],\{\boldsymbol{i},\mathbf{1},\mathbf{3}\},\{\boldsymbol{j},\mathbf{1},\mathbf{3}\}];$

$\boldsymbol{M}\mathbf{//MatrixForm}$

$\mathbf{Tr}[\boldsymbol{M}]$

$$\begin{pmatrix} a[1,1] & a[1,2] & a[1,3] \\ a[2,1] & a[2,2] & a[2,3] \\ a[3,1] & a[3,2] & a[3,3] \end{pmatrix}$$

a[1,1]+a[2,2]+a[3,3]

Матрицу можно транспонировать

$\mathbf{MatrixForm}[\mathbf{Transpose}[\{\{\boldsymbol{a},\boldsymbol{b}\},\{\boldsymbol{c},\boldsymbol{d}\}\}]]$

$$\begin{pmatrix} a & c \\ b & d \end{pmatrix}$$

Можно найти обратную матрицу (если она существует).

$\boldsymbol{A}=\mathbf{Inverse}\begin{bmatrix}\mathbf{1} & \mathbf{2}\\ \mathbf{3} & \mathbf{4}\end{bmatrix}$

$$\{\{-2,1\},\{\frac{3}{2},-\frac{1}{2}\}\}$$

$\mathbf{MatrixForm}[\boldsymbol{A}]$

$$\begin{pmatrix} -2 & 1 \\ \frac{3}{2} & -\frac{1}{2} \end{pmatrix}$$

Аналогично

$\mathbf{Inverse}\left[\begin{pmatrix}\boldsymbol{a} & \boldsymbol{b}\\ \boldsymbol{c} & \boldsymbol{d}\end{pmatrix}\right]\mathbf{//MatrixForm}$

$$\begin{pmatrix} \frac{d}{-bc+ad} & -\frac{b}{-bc+ad} \\ -\frac{c}{-bc+ad} & \frac{a}{-bc+ad} \end{pmatrix}$$

Имеются функции для автоматического создания матриц некоторых типов. На сомом деле такие функции создают список списков, который мы можем интерпретировать как матрицу. Функция `IdentityMatrix` создает единичную матрицу указанного размера. Функция `DiagonalMatrix` создает диагональную матрицу. Функция `Array[f,{n1,n2}]` генерирует $n_1 \times n_2$ массив – двойной список с элементами `f[i1,i2]`. Здесь имя `f` интерпретируется как имя функции, например, `Sin`. Вот несколько примеров.

$\mathbf{MatrixForm}[\mathbf{IdentityMatrix}[\mathbf{3}]]$

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

$\mathbf{MatrixForm}[\mathbf{DiagonalMatrix}[\{\boldsymbol{a},\boldsymbol{b},\boldsymbol{c}\}]]$

$$\begin{pmatrix} a & 0 & 0 \\ 0 & b & 0 \\ 0 & 0 & c \end{pmatrix}$$

MatrixForm[**Array**[$\boldsymbol{a}$,{**3**,**3**}]]

$$\begin{pmatrix} a[1,1] & a[1,2] & a[1,3] \\ a[2,1] & a[2,2] & a[2,3] \\ a[3,1] & a[3,2] & a[3,3] \end{pmatrix}$$

или, например,

$\boldsymbol{f}$[**x_**,**y_**]:= $\boldsymbol{x\,y}$

Array[$\boldsymbol{f}$,{**3**,**3**}]//**MatrixForm**

$$\begin{pmatrix} 1 & 2 & 3 \\ 2 & 4 & 6 \\ 3 & 6 & 9 \end{pmatrix}$$

Здесь мы создали функцию двух переменных *f(x,y)* и с ее помощью генерировали матрицу размера 3x3.

Функции `Eigenvalues` и `Eigenvectors` вычисляют собственные числа и собственные вектора матрицы.

Eigenvalues[$\begin{pmatrix} \mathbf{1} & \mathbf{4} \\ \mathbf{2} & \mathbf{3} \end{pmatrix}$]

{-1,5}

Eigenvectors[$\begin{pmatrix} \mathbf{1} & \mathbf{4} \\ \mathbf{2} & \mathbf{3} \end{pmatrix}$]

{{-2,1},{1,1}}

Напомним, что скалярное произведение векторов вычисляется с использованием операции «точка».

{1,2}.{3,4}

11

Векторное произведение векторов вычисляется функцией `Cross`.

Cross[{1,2,3},{2,4,5}]

```
{-2,1,0}
```

Cross[{1,2,3,2},{2,4,5,1},{1,1,1,1}]

```
{-3,5,-3,1}
```

Векторное произведение `n` векторов в `n+1` мерном пространстве даст вектор ортогональный ко всем сомножителям. Ортогональность можно проверить с помощью скалярного произведения результирующего вектора на каждый вектор сомножитель.

При решении систем линейных уравнений полезной является функция приведения матрицы к диагональному виду (если это возможно). Функция `RowReduce[Matr]` выполняет приведение матрицы к ступенчатой форме методом Гаусса. При этом невырожденная матрица будет приводиться к единичной.

RowReduce$\left[\begin{pmatrix} \mathbf{1} & \mathbf{2} & \mathbf{3} \\ \mathbf{4} & \mathbf{5} & \mathbf{6} \\ \mathbf{7} & \mathbf{8} & \mathbf{8} \end{pmatrix}\right]$//**MatrixForm**

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

$$\mathbf{RowReduce}\left[\begin{pmatrix}\mathbf{1} & \mathbf{2} & \mathbf{3}\\ \mathbf{4} & \mathbf{5} & \mathbf{6}\end{pmatrix}\right]\mathbf{//MatrixForm}$$

$$\begin{pmatrix}1 & 0 & -1\\ 0 & 1 & 2\end{pmatrix}$$

$$\mathbf{RowReduce}\left[\begin{pmatrix}\mathbf{1} & \mathbf{2} & \mathbf{3}\\ \mathbf{4} & \mathbf{5} & \mathbf{6}\\ \mathbf{7} & \mathbf{8} & \mathbf{9}\end{pmatrix}\right]\mathbf{//MatrixForm}$$

$$\begin{pmatrix}1 & 0 & -1\\ 0 & 1 & 2\\ 0 & 0 & 0\end{pmatrix}$$

В последнем примере у ненулевой вырожденной матрицы последняя строка приводится к нулевой, а последний столбец будет ненулевым (если он не был нулевым с самого начала).

1.3 Алгебраические преобразования

Свое название «Системы аналитических вычислений» получили в связи их умением выполнять различные преобразования алгебраических выражений. Для того, чтобы выполнить какое-либо преобразование выражения, его (выражение) надо передать аргументом соответствующей функции. В этом параграфе мы рассмотрим примеры таких функций.

Функция `Factor` раскладывает алгебраическое выражение на множители.

$$\mathbf{Factor}[\boldsymbol{a^2 + 2ab + b^2}]$$

$$(a+b)^2$$

или

$$\mathbf{Factor}[\mathbf{6 + 11x - 3x^2 - 2x^3}]$$

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

Можно ли x^4+4 разложить на множители без комплексных чисел. Давайте посмотрим, что скажет Mathematica:

$$\mathbf{Factor}[\boldsymbol{x^4 + 4}]$$

$$(2-2x+x^2)(2+2x+x^2)$$

Если мы хотим разрешить комплексные числа при разложении на множители, то мы можем использовать следующий вариант:

$$\mathbf{Factor}[\boldsymbol{x^2 + 9}, \mathbf{GaussianIntegers \to True}]$$

$$(-3i + x)(3i + x)$$

`GaussianIntegers->True` - это так называемая опция (необязательный аргумент функции): «Я хочу разрешить комплексные числа с целыми коэффициентами в результате».

По умолчанию *Mathematica* раскладывает на множители с целыми коэффициентами, поэтому следующий пример не раскладывается.

$$\mathbf{Factor}[\boldsymbol{x^2 - 3}]$$

Mathematica не раскладывает на множители «в радикалах». Разложение на множители «в радикалах» - это не разложение в обычном смысле слова; это больше похоже на отыскание корней уравнений, которое делается командой `Solve` (см. следующий параграф):

$\mathbf{Solve}[\boldsymbol{x^2 - 3 == 0, x}]$

$\{\{x \to -\sqrt{3}\}, \{x \to \sqrt{3}\}\}$

Однако, функция `Factor` имеет опцию `Extension`, использование которой изменяет поведение алгоритма разложения на множители.

$\mathbf{Factor}[\boldsymbol{1 + x^4}]$

$1 + x^4$

$\mathbf{Factor}[\boldsymbol{1 + x^4}, \mathbf{Extension} \to \sqrt{\mathbf{2}}]$

$-(-1 + \sqrt{2}x - x^2)(1 + \sqrt{2}x + x^2)$

$\mathbf{Factor}[\boldsymbol{1 + x^4}, \mathbf{Extension} -> \{\sqrt{\mathbf{2}}, \boldsymbol{I}\}]$

$\frac{1}{4}(\sqrt{2} - (1 + i)x)(\sqrt{2} - (1 - i)x)(\sqrt{2} + (1 - i)x)(\sqrt{2} + (1 + i)x)$

$\mathbf{Factor}[\boldsymbol{2 + 2\sqrt{2}x + x^2}]$

$2 + 2\sqrt{2}x + x^2$

$\mathbf{Factor}[\boldsymbol{2 + 2\sqrt{2}x + x^2}, \mathbf{Extension} -> \mathit{Automatic}]$

$(\sqrt{2} + x)^2$

Использование опции `GaussianIntegers→True` эквивалентно использованию опции `Extension→I`.

$\mathbf{Factor}[\boldsymbol{1 + x^2}, \mathbf{GaussianIntegers} -> \mathit{True}]$

$(-i + x)(i + x)$

$\mathbf{Factor}[\boldsymbol{1 + x^2}, \mathbf{Extension} -> \boldsymbol{I}]$

$(-i + x)(i + x)$

Есть еще одна полезная опция `Trig→True`, которая позволяет разложить в произведение (если это возможно) тригонометрические выражения.

$\mathbf{Factor}[\mathbf{Sin}[\boldsymbol{2x}] + \mathbf{Sin}[\boldsymbol{2y}], \mathbf{Trig} \to \mathbf{True}]$

$2\mathrm{Cos}[x - y]\mathrm{Sin}[x + y]$

Если вы раскладываете на множители полиномы, включающие дроби, то *Mathematica* приводит все к общему знаменателю

$\mathbf{Factor}\left[\boldsymbol{x^2 - \frac{4}{9}}\right]$

$\frac{1}{9}(-2 + 3x)(2 + 3x)$

Mathematica может раскладывать на множители выражения, содержащие функции вместо простых переменных:

$\mathbf{Factor}[\boldsymbol{f[x]^2 - g[x]^2}]$

$(f[x] - g[x])(f[x] + g[x])$

Функция `Expand` обратная к функции `Factor`. Она умеет раскрывать скобки в алгебраическом выражении.

$\mathbf{Expand}[\boldsymbol{(a + b)^5}]$

$a^5 + 5a^4b + 10a^3b^2 + 10a^2b^3 + 5ab^4 + b^5$

В формате `Expand[expr, patt]` функция оставляет не разложенной любую часть выражения `expr`, которая не содержит выражения `patt`.

$\mathbf{Expand}[(\boldsymbol{a}+\boldsymbol{b})^{\mathbf{2}}(\mathbf{1}+\boldsymbol{x})^{\mathbf{2}}]$

$a^2+2ab+b^2+2a^2x+4abx+2b^2x+a^2x^2+2abx^2+b^2x^2$

$\mathbf{Expand}[(\boldsymbol{a}+\boldsymbol{b})^{\mathbf{2}}(\mathbf{1}+\boldsymbol{x})^{\mathbf{2}},\boldsymbol{x}]$

$(a+b)^2+2(a+b)^2x+(a+b)^2x^2$

В следующем примере раскрываются скобки только у выражения вида `1+x`

$\mathbf{Expand}[(\mathbf{1}+\boldsymbol{x})^{\mathbf{3}}+(\mathbf{2}+\boldsymbol{x})^{\mathbf{4}}+(\mathbf{1}+\boldsymbol{y})^{\mathbf{2}},\mathbf{1}+\boldsymbol{x}]$

$1+3x+3x^2+x^3+(2+x)^4+(1+y)^2$

В следующем примере остаются нетронутыми элементы выражения, которые не соответствуют шаблону `x[_]`

$\mathbf{Expand}[(\boldsymbol{a}+\boldsymbol{b})(\boldsymbol{x}[\mathbf{1}]+\boldsymbol{x}[\mathbf{2}])^{\mathbf{2}},\boldsymbol{x}[\_]]$

$(a+b)x[1]^2+2(a+b)x[1]x[2]+(a+b)x[2]^2$

Опция `Trig→True` подсказывает, что нужно выполнять тригонометрические разложения

$\mathbf{Expand}[\mathbf{Sin}[\boldsymbol{x}+\boldsymbol{y}],\mathbf{Trig}\rightarrow\mathbf{True}]$

$\mathrm{Cos}[y]\mathrm{Sin}[x]+\mathrm{Cos}[x]\mathrm{Sin}[y]$

Без этой опции разложение не выполняется

$\mathbf{Expand}[\mathbf{Sin}[\boldsymbol{x}+\boldsymbol{y}]]$

$\mathrm{Sin}[x+y]$

Функция `Together` выполняет приведение к общему знаменателю:

$\mathbf{Together}\left[\frac{\boldsymbol{a}}{\boldsymbol{b}}+\frac{\boldsymbol{c}}{\boldsymbol{d}}\right]$

$$\frac{b\,c+a\,d}{b\,d}$$

Наоборот, функция `Apart` выполняет разложение дробей:

$\mathbf{Apart}\left[\frac{\mathbf{b\,c+a\,d}}{\mathbf{b\,d}}\right]$

$$\frac{a}{b}+\frac{c}{d}$$

В выражении собирание коэффициентов при одинаковых степенях переменной, определяемой вторым аргументом, можно выполнить с помощью функции `Collect`:

$\mathbf{Collect}[\boldsymbol{a}\,\boldsymbol{x}+\boldsymbol{b}\,\boldsymbol{x}+\boldsymbol{c}\,\boldsymbol{y}+\boldsymbol{d}\,\boldsymbol{y},\boldsymbol{x}]$

$(a+b)x+cy+dy$

и

$\mathbf{Collect}[\boldsymbol{a}\,\boldsymbol{x}+\boldsymbol{b}\,\boldsymbol{x}+\boldsymbol{c}\,\boldsymbol{y}+\boldsymbol{d}\,\boldsymbol{y},\{\boldsymbol{x},\boldsymbol{y}\}]$

$(a+b)x+(c+d)y$

Функции `TrigExpand` и `TrigReduce` выполняют действия, аналогичные `Expand` и `Collect` применительно к тригонометрическим выражениям.

$\mathbf{TrigExpand}[\mathbf{Sin}[\mathbf{3}\boldsymbol{x}]]$

$3\mathrm{Cos}[x]^2\mathrm{Sin}[x]-\mathrm{Sin}[x]^3$

$\mathbf{TrigReduce}[\mathbf{Sin}[\boldsymbol{x}]\mathbf{Cos}[\mathbf{2}\boldsymbol{y}]]$

$\frac{1}{2}(\mathrm{Sin}[x-2y]+\mathrm{Sin}[x+2y])$

Функция TrigToExp преобразует тригонометрические и гиперболические функции в экспоненты

TrigToExp[**Sin**[$2x$]]

$$\frac{1}{2}ie^{-2ix} - \frac{1}{2}ie^{2ix}$$

TrigToExp[**ArcTanh**[x]]

$$-\frac{1}{2}\text{Log}[1-x] + \frac{1}{2}\text{Log}[1+x]$$

Функция ExpToTrig выполняет действия, обратные TrigToExp.

ExpToTrig[**Exp**[Ix]]

$$\text{Cos}[x] + i\text{Sin}[x]$$

ExpToTrig[**Sqrt**[I]]

$$\frac{1+i}{\sqrt{2}}$$

Упрощение алгебраических выражений выполняется функцией Simplify.

Simplify $\left[\frac{x^2+2xy+y^2}{x+y}\right]$

x+y

Simplify[expr] и FullSimplify[expr] – приводят выражение expr к наиболее простому виду. Simplify пробует найти простейшую форму выражения. Часто бывает не совсем ясно, какая из форм будет проще, и разные люди могут иметь различные мнения о простоте. *Mathematica* считает простейшим выражение с наименьшим количеством элементов.

Simplify[**Sin**[x]^2 + **Cos**[x]^2]

1

Однако часто она не может ничего сделать с выражением и поэтому вам надо вызывать самому функции преобразования (Factor, Expand и др.), чтобы выполнить упрощение.

Функция FullSimplify умеет выполнять упрощения со многими специальными функциями.

Для выполнения упрощений с учетом дополнительных условий используется второй аргумент этих функций

Simplify[**Sqrt**[x^2], **Element**[x, **Reals**]]

Abs[x]

То же можно выполнить командой

Simplify[**Sqrt**[x^2], $x \in$ **Reals**]

Здесь символ ∈ представляет инфиксную (между аргументами) запись функции двух переменных Element[x,Reals].

Вот пример вызова функции Simplify без каких – либо предположений

Simplify[**Sqrt**[x^2]]

$$\sqrt{x^2}$$

Тогда

Simplify[**Sqrt**[x^**2**], $x > 0$]
x
или
Simplify[**Sqrt**[x^**2**], $x \in$ **Reals**]
Abs[x]
Аналогично
Simplify[**Sqrt**[$x^{**2**}y$^**2**], {$x \in$ **Reals**, $y < 0$}]
$-y$Abs[x]
Второй аргумент можно записывать, используя опцию `Assumptions`.
Simplify[**Sqrt**[x^**2**]/x + **Sqrt**[y^**2**]/y, **Assumptions** → {$x > 0, y > 0$}]
2

1.4 Решение алгебраических уравнений

Большинство алгебраических преобразований мы можем выполнить самостоятельно. Однако решение уравнений является боле сложной задачей. И в этом вопросе помощь *Mathematica* может быть значительной. Здесь мы рассматриваем функции пакета, с помощью которых можно решать алгебраические уравнения. В системе реализованы все известные методы «точного» определения корней уравнений. Но даже в тех случаях, когда соответствующих формул нет, мы можем найти приближенное значение корня (корней).

Для символьного решения уравнения используется функция `Solve`

Solve[$x^2 - 2x == 0, x$]
{{$x \to 0$}, {$x \to 2$}}

Первый аргумент - это уравнение, которое должно быть решено. Второй аргумент - переменная, относительно которой мы решаем уравнение. Результат возвращается в форме списка правил подстановки («замены») с использованием операции '→' стрелка (минус и знак больше). В данном случае показано, что решение *x* равно 0 или 2. Вот пример решения кубического уравнения

$e = x^3 - 2x + 1 == 0$
Solve[e, x]

$$\left\{\{x \to 1\}, \left\{x \to \frac{1}{2}\left(-1 - \sqrt{5}\right)\right\}, \left\{x \to \frac{1}{2}\left(-1 + \sqrt{5}\right)\right\}\right\}$$

Помните, что имена неизвестных, фигурирующих в уравнениях, не должны содержать значений. Если, например, переменной `x`, используемой в примере, ранее было присвоено значение, то его (значение) надо «забыть». Попробуйте, например, выполнить команды

$x = 5$;
Solve[$x^2 - 3x + 2 == 0, x$]
```
Solve::ivar: _5_ is not a valid variable>>
```
Solve[False, 5]

Вы получаете сообщение об ошибке и результат Solve[False,5]. Что произошло? Перед решением алгебраического уравнения вы присвоили переменной *x*

значение 5. Система подставила в выражение $x^2 - 3x + 2$ вместо *x* значение 5, получила значение 12, которое не совпадает с 0. В результате, вместо ожидаемого уравнения $x^2 - 3x + 2 == 0$ первый аргумент функции `Solve` равен False.

Чтобы такого не происходило, искомое имя (в данном случае *x*) не должно содержать значения. Для этого перед вызовом функции, решающей уравнение или систему уравнений, следует вызвать функцию `Remove[x]` или `Clear[x]`. Функциям `Remove` и `Clear` можно передавать имена нескольких переменных, перечисляемых через запятую.

Замечание по «очистке» имен переменных касается всех примеров этого параграфа. Если перед кодом примера вы будете добавлять команду `Remove[имя1, имя2, ...]`, где `имя1, имя2,` это имена переменных, используемых в коде, то в большинстве случаев этого достаточно для корректного выполнения текущего примера.

Иное по сравнению с `Solve` представление решения дает функция `Roots`

$\mathbf{rts = Roots[}e, x\mathbf{]}$

$x == \frac{1}{2}\left(-1 - \sqrt{5}\right) \;||\; x == \frac{1}{2}\left(-1 + \sqrt{5}\right) \;||\; x == 1$

Второе представление можно преобразовать к первому с помощью функции `ToRules`

$\mathbf{ToRules[rts]}$

$\text{Sequence}[\{x \to \frac{1}{2}(-1 - \sqrt{5})\}, \{x \to \frac{1}{2}(-1 + \sqrt{5})\}, \{x \to 1\}]$

Если уравнения содержат другие переменные, то они трактуются как константы

$\mathbf{Solve}[x^2 - 5xy + 4y^2 == 0, x]$

$\{\{x \to y\}, \{x \to 4y\}\}$

Вы можете решать системы уравнений, содержащие более чем одну неизвестную, используя для этого список уравнений и список неизвестных

$\mathbf{Solve}[\{2x + 3y == 13, 3x - 2y == 0\}, \{x, y\}]$

$\{\{x \to 2, y \to 3\}\}$

Первый аргумент - список решаемых уравнений. Второй аргумент - список переменных, относительно которых решаются уравнения.

Приведенная выше система имеет одно решение (одна пара значений *x* и *y*). Если существует более одного решения, то будет получен список списков значений

$\mathbf{Solve}[\{x^2 + y^2 == 16, x^2 - 4 == y\}, \{x, y\}]$

$\{\{x \to 0, y \to -4\}, \{x \to -\sqrt{7}, y \to 3\}, \{x \to \sqrt{7}, y \to 3\}\}$

Первый элемент результата $\{x \to 0, y \to -4\}$ - это первое решение и т.д. Добавляя команду `//TableForm` к команде `Solve`, мы делаем вывод более читабельным

$\mathbf{Solve}[\{x^2 + y^2 == 16, x^2 - 4 == y\}, \{x, y\}]\mathbf{//TableForm}$

$x \to 0 \qquad y \to -4$

$x \to -\sqrt{7} \qquad y \to 3$

$x \to \sqrt{7} \qquad y \to 3$

Каждая строка представляет одно решение. Обратите внимание на кратный корень.

Если решения не существует, то возвращается пустой список:

$\mathbf{Solve}[\{x^2 - 4 == 0, x^2 - 3 == 0\}, x]$

{ }

Решение уравнений функцией Solve возвращается в форме списка правил подстановки. Поясним эти правила на примере.

$x^2/.\, x \to 5$

25

Здесь сказано, что надо заменить *x* на 5 в выражении x^2. Оператор /. (слеш и точка) читается «заменить» и → (стрелка) читается как «на». Все выражение следует читать «в x^2 заменить *x* на 5».

Оператор /. (слеш и точка) является постфиксной формой функции `ReplaceAll`

$\mathbf{ReplaceAll}[x^2, x->5]$

Можно использовать список правил замены более чем одной переменной за один раз

$x^2 + y^2/.\{x \to 5, y \to 10\}$

125

Когда вы используете список списков правил замены, вы получаете список результатов

$x^2/.\{\{x \to 5\}, \{x \to 6\}\}$

```
{25,36}
```

или

$x^2 + y^2/.\{\{x \to 5, y \to 10\}, \{x \to 6, y \to 10\}\}$

```
{125,136}
```

Вернемся к первому примеру решения уравнения

$s = \mathbf{Solve}[x^2 - 2x == 0, x]$

$\{\{x \to 0\}, \{x \to 2\}\}$

Мы получили решение в виде списка списков правил подстановки, и можем использовать его для замены значений в любых выражениях. Например, мы можем проверить, что ответ правильный, подставляя решение в исходное уравнение

$x^2 - 2x/.\, s$

{0, 0}

Список из двух нулей означает, что оба решения дают нулевые значения, когда мы подставляем их в исходное выражение.

$x^2 - 3x/.\, s[[2]]$

-2

т.к. `s[[2]]={x->2}`, и

$x/.s[[2]]$
2
Можно прямо использовать команду `Solve` для замены в выражении
$x^2+1/.\mathrm{Solve}[x^2-2x==0,x]$
`{1,5}`
или
$x/.\mathrm{Solve}[x^3+ax^2+ax+1==0,x][[2]]$
$\frac{1}{2}(1-a-\sqrt{-3-2a+a^2})$
Запись `Solve[x^3+a x^2 + a x+1==0,x][[2]]` означает выбор второго элемента списка правил подстановки, т.е. второго корня уравнения.

Для получения списка значений решений удобно использовать следующую форму
$x/.\mathrm{Solve}[x^2-4x+3==0,x]$
`{1,3}`
Некоторые уравнения (например, полиномы 5 степени и выше) не могут быть решены в явном виде
$s=\mathrm{Solve}[x^6+x^5+x^2+1==0,x]$
$\{\{x\to \mathrm{Root}[1+\#1^2+\#1^5+\#1^6\&,1]\},$
$\{x\to \mathrm{Root}[1+\#1^2+\#1^5+\#1^6\&,2]\},$
$\{x\to \mathrm{Root}[1+\#1^2+\#1^5+\#1^6\&,3]\},$
$\{x\to \mathrm{Root}[1+\#1^2+\#1^5+\#1^6\&,4]\},$
$\{x\to \mathrm{Root}[1+\#1^2+\#1^5+\#1^6\&,5]\},$
$\{x\to \mathrm{Root}[1+\#1^2+\#1^5+\#1^6\&,6]\}\}$
Здесь функция Root[f,k] представляет «точное значение» k – го корня полиномиального уравнения f[x]==0. Под «точным значением» понимается величина, которая может быть вычислена с любой степенью точности. Например
$N[s[[1]]]$
$\{x\to -1.1540824-0.613722i\}$
или
$N[s[[1]],20]$
$\{x\to -1.1540824762585313148-0.61372296835498784283i\}$
Фактически, мы сразу могли написать
$s=N[\mathrm{Solve}[x^6+x^5+x^2+1==0,x]]$
и получить *приближенные* значения всех шести корней. В то же время, функция Root[1 + #1^2 + #1^5 + #1^6&, k] представляет «точное» значение.

Применение функции `N[Solve[…]]` эквивалентно вызову функции `NSolve[…]` с теми же аргументами. Функция `NSolve` находит приближенно все корни алгебраического (полиномиального) уравнения.
$\mathrm{NSolve}[1+2x+3x\text{^}2+4x\text{^}3==0,x]$
$\{\{x\to -0.60583\},\{x\to -0.07208-0.6383i\},\{x\to -0.07208+0.6383i\}\}$

Функция `NRoots` находит приближенные решения полиномиальных уравнений, но ответ представляет по – другому. Например, решение предыдущей задачи она представляет в виде уравнений

NRoots[**1** + **2***x* + **3***x*^**2** + **4***x*^**3** == **0**, *x*]

$x == -0.60583 \;||\; x == -0.0721 - 0.6383\,i \;||\; x == -0.0721 + 0.6383\,i$

При желании преобразовать последнее решение в список правил подстановки можно выполнить команду

{**ToRules**[%]}

$\{\{x \to -0.60583\}, \{x \to -0.07208 - 0.6383i\}, \{x \to -0.07208 + 0.6383i\}\}$

Если уравнение содержит параметр, то его корень будет зависеть от этого параметра. Тогда для построения функции (значение корня в зависимости от параметра) можно поступить так

$s = \mathbf{First}[\mathbf{Solve}[x^2 - 2x - a^4 == 0, x]]$

$z[\mathrm{t\_}] = (x/.s)/.a \to t$

$\{z[x], z'[x], z[3], z'[3]\}$

$$\{1 - \sqrt{1 + x^4}, -\frac{2x^3}{\sqrt{1 + x^4}}, 1 - \sqrt{82}, -27\sqrt{\frac{2}{41}}\}$$

Построенная в этом примере функция *z*[*t*] является «настоящей» функцией – мы вычислили ее значение в точке и взяли производную. В этом примере задание функции с использованием отложенного присваивания в виде

$z[\mathrm{a\_}] := x/.s$

некорректно, т.к. уже команда

z[x]

дает неверный ответ. Но сработает непосредственное присваивание

Remove[*x*, *s*, *z*, *a*, *t*]

$s = \mathbf{First}[\mathbf{Solve}[x^2 - 2x - a^4 == 0, x]]$

$z[\mathrm{a\_}] = x/.s$

$\{z[x], z'[x], z[3], z'[3]\}$

Если нужно использовать отложенное присваивание, то можно использовать функцию `Evaluate`

$z[\mathrm{a\_}] := \mathbf{Evaluate}[x/.s]$

Как мы видели, функция `Solve` позволяет находить все корни уравнения или системы уравнений (если это возможно).

Для решения линейных систем уравнений можно использовать функцию `LinearSolve`. Функция `LinearSolve[M,b]` находит вектор *x*, который удовлетворяет матричному уравнению `M·x=b`, где `M` – матрица коэффициентов, `b` – вектор свободных членов.

$$\mathbf{LinearSolve}\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}, \begin{bmatrix} 2 \\ 3 \end{bmatrix}$$

$\{\{-1\}, \{\frac{3}{2}\}\}$

Если решение не единственно, то возвращается только одно решение.

$\boldsymbol{s} = \mathbf{LinearSolve}[\boldsymbol{m} = \begin{pmatrix} \mathbf{1} & \mathbf{2} & \mathbf{3} \\ \mathbf{2} & \mathbf{3} & \mathbf{4} \end{pmatrix}, \boldsymbol{v} = \{-\mathbf{1}, \mathbf{3}\}]$

```
{9,-5,0}
```

Заметим, что присваивания $m = \begin{pmatrix} 1 & 2 & 3 \\ 2 & 3 & 4 \end{pmatrix}$ и $v = \{-1,3\}$, использованные в предыдущем примере, являются операциями, возвращающими результат. Результатом операций присваивания являются выражения стоящие в правой части, т.е. матрица $\begin{pmatrix} 1 & 2 & 3 \\ 2 & 3 & 4 \end{pmatrix}$ и список $\{-1,3\}$. Т.о. за одно обращение к функции `LinearSolve` мы выполнили присвоение переменных *m* и *v* , а также корректную передачу параметров.

В этом примере мы должны были бы получить бесконечное множество решений. Для получения общего решения в подобных случаях надо использовать функцию `Solve`.

$\mathbf{Solve}[\boldsymbol{m}.\{\boldsymbol{a},\boldsymbol{b},\boldsymbol{c}\} == \boldsymbol{v}, \{\boldsymbol{a},\boldsymbol{b},\boldsymbol{c}\}]$

```
Solve::svars: Equations may not give solutions for all
"solve" variables.
```

$\{\{b \to 13 - 2a, c \to -9 + a\}\}$

При этом мы получили предупреждающее сообщение о том, что функция `Solve` не уверена, что смогла определить все решения системы. Обратите также внимание на способ записи системы уравнений в этом примере. Она имеет вид

$\boldsymbol{m}.\{\boldsymbol{a},\boldsymbol{b},\boldsymbol{c}\} == \boldsymbol{v}$

$\{a + 2b + 3c, 2a + 3b + 4c\} == \{-1,3\}$

Если решение системы не существует, то выводится соответствующее сообщение, а также повторяется ввод.

$\mathbf{LinearSolve}[\begin{pmatrix} \mathbf{1} & \mathbf{2} \\ \mathbf{3} & \mathbf{4} \\ \mathbf{5} & \mathbf{6} \end{pmatrix}, \{-\mathbf{1}, \mathbf{2}, -\mathbf{3}\}]$

```
LinearSolve::nosol: Linear equation encountered that has
no solution.
```

$\text{LinearSolve}[\{\{1,2\},\{3,4\},\{5,6\}\},\{-1,2,-3\}]$

При решении системы с использованием функции `LinearSolve[M,b]` аргумент b может быть вектором или матрицей. Если он матрица, то определяются решения, соответствующие каждому столбцу матрицы b.

$\boldsymbol{s} = \mathbf{LinearSolve}[\boldsymbol{m} = \begin{pmatrix} \mathbf{1} & \mathbf{2} & \mathbf{3} \\ \mathbf{2} & \mathbf{3} & \mathbf{4} \\ \mathbf{3} & \mathbf{1} & \mathbf{3} \end{pmatrix}, \boldsymbol{v} = \begin{pmatrix} \mathbf{7} & \mathbf{1} \\ \mathbf{12} & \mathbf{2} \\ \mathbf{7} & \mathbf{3} \end{pmatrix}]$

```
{{2,1},{4,0},{-1,0}}
```

Здесь первое решение дается тройкой чисел `{2,4,-1}`, а второе – тройкой `{1,0,0}`.

Далеко не всегда функции *Mathematica* могут найти решение. Иногда им надо помочь, выполнив предварительно некоторые преобразования уравнения или системы уравнений.

Функция `Reduce[eqns,vars]` упрощает уравнения, заданные списком `eqns`, и пытается решить их относительно переменных, заданных списком `vars`. Уравнения, возвращаемые функцией `Reduce`, эквивалентны первоначальным и содержат все возможные решения. Результирующие уравнения объединяются с использованием операций логического ‘И’ (&&) или логического ‘ИЛИ’ (||).

$\mathbf{Reduce}[\{\boldsymbol{x}+\boldsymbol{y}+\boldsymbol{z}==\mathbf{1},\boldsymbol{x}-\boldsymbol{y}-\boldsymbol{z}==\mathbf{0}\},\{\boldsymbol{x}\}]$

$y==\frac{1}{2}-z\ \&\&\ x==\frac{1}{2}$

$\mathbf{Reduce}[\{\boldsymbol{x}+\boldsymbol{y}+\boldsymbol{z}==\mathbf{1},\boldsymbol{x}-\boldsymbol{y}-\boldsymbol{z}==\mathbf{0}\},\{\boldsymbol{x},\boldsymbol{z}\}]$

$x==\frac{1}{2}\ \&\&\ z==\frac{1}{2}-y$

$\mathbf{Reduce}[\{\boldsymbol{a}\boldsymbol{x}+\boldsymbol{y}+\boldsymbol{z}==\mathbf{1},\boldsymbol{x}-\boldsymbol{y}-\boldsymbol{z}==\mathbf{0}\},\{\boldsymbol{x},\boldsymbol{y}\}]$

$1+a\neq 0\ \&\&\ x==\frac{1}{1+a}\ \&\&\ y==x-z$

В формате вызова `Reduce[eqns,vars,elims]` функция упрощает уравнения и пытается исключить неизвестные `elims`.

$\mathbf{Reduce}[\{\boldsymbol{a}\,\boldsymbol{x}+\boldsymbol{y}+\boldsymbol{z}==\mathbf{1},\boldsymbol{x}-\boldsymbol{y}-\boldsymbol{z}==\mathbf{0}\},\{\boldsymbol{y}\},\{\boldsymbol{a}\}]$

$y==x-z\ \&\&\ x\neq 0$

(решаем относительно *y* и исключаем *a*)

Функция `Eliminate[eqns,vars]` упрощает систему уравнений путем исключения неизвестных `vars`. Она работает хорошо с линейными и полиномиальными уравнениями.

$\mathbf{Eliminate}[\{\boldsymbol{x}^{\mathbf{2}}==\mathbf{2}+\mathbf{2}\boldsymbol{x}+\boldsymbol{y},\boldsymbol{x}+\boldsymbol{y}==\boldsymbol{z}\},\boldsymbol{y}]$

$z==-2-x+x^2$

Здесь переменная *y* исключена.

$\mathbf{Eliminate}[\{\boldsymbol{a}\boldsymbol{x}^{\mathbf{2}}+\boldsymbol{b}\boldsymbol{x}\boldsymbol{y}+\boldsymbol{c}\boldsymbol{y}^{\mathbf{2}}==\mathbf{0},\boldsymbol{x}+\boldsymbol{y}==\mathbf{0}\},\boldsymbol{x}]$

$cy^2==(-a+b)y^2$

Здесь исключена переменная *x*. В результате получено уравнение относительно неизвестной y.

Функция `SolveAlways[eqns, vars]` возвращает список значений параметров, подстановка которых превращает уравнение в тождество относительно переменных `vars`.

$\mathbf{SolveAlways}[\boldsymbol{a}\,\boldsymbol{x}+\boldsymbol{b}==\mathbf{0},\boldsymbol{x}]$

$\{\{a\to 0,b\to 0\}\}$

$\mathbf{SolveAlways}[\boldsymbol{a}\boldsymbol{x}^{\mathbf{3}}+\boldsymbol{x}(\boldsymbol{x}-\boldsymbol{b})^{\mathbf{2}}+\boldsymbol{c}\boldsymbol{x}==\mathbf{0},\boldsymbol{x}]$

$\{\{a\to -1,c\to 0,b\to 0\}\}$

$\mathbf{SolveAlways}[\boldsymbol{a}\boldsymbol{x}^{\mathbf{3}}+(\boldsymbol{x}-\boldsymbol{b})^{\mathbf{2}}+\boldsymbol{c}==\mathbf{0},\boldsymbol{x}]$

$\{\ \}$

Используя эту функцию, можно вывести теорему Виета для полиномов 2 – й и 3 – й степени.

$\mathbf{SolveAlways}[\boldsymbol{x}^{\mathbf{2}}+\boldsymbol{a}\boldsymbol{x}+\boldsymbol{b}==(\boldsymbol{x}-\mathbf{x1})(\boldsymbol{x}-\mathbf{x2}),\boldsymbol{x}]$

$\{\{a\to -\mathrm{x1}-\mathrm{x2},b\to \mathrm{x1x2}\}\}$

$\mathbf{SolveAlways}[\boldsymbol{x}^{\mathbf{3}}+\boldsymbol{b}\boldsymbol{x}^{\mathbf{2}}+\boldsymbol{c}\boldsymbol{x}+\boldsymbol{d}==(\boldsymbol{x}-\mathbf{x1})(\boldsymbol{x}-\mathbf{x2})(\boldsymbol{x}-\mathbf{x3}),\boldsymbol{x}]$

$\{\{b \to -x1 - x2 - x3, c \to x1x2 + x1x3 + x2x3, d \to -x1x2x3\}\}$

Функция `FindRoot[eqn, {x,x0}]` пытается найти численное решение уравнения `eqn` методом Ньютона, используя начальное приближение `x=x0`.

$\mathbf{FindRoot}[x^2 == \mathbf{Exp}[x+2], \{x, -1\}]$

$\{x \to -1.37015\}$

$\mathbf{FindRoot}[x^5 + 3e^{-x^4} == 0, \{x, -1\}, \mathbf{WorkingPrecision} \to 40]$

$\{x \to -1.010909816596691797197753835847306190493\}$

Помните, что определяется один ближайший к начальному приближению корень.

$\mathbf{FindRoot}[\mathbf{Cos}[x]^4 + 2\mathbf{Cos}[x]^2 == 0, \{x, 0.1\}]$

$\{x \to 4.71239\}$

Выбирая другое начальное приближение, мы получаем другой корень

$\mathbf{FindRoot}[\mathbf{Cos}[x]^4 + 2\mathbf{Cos}[x]^2 == 0, \{x, -1\}]$

$\{x \to -1.5708\}$

Вот что происходит, если функция FindRoot не может найти решение.

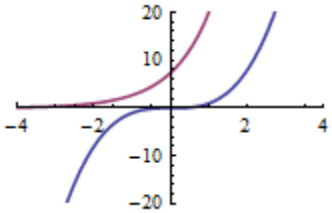
$\mathbf{FindRoot}[x^3 == \mathbf{Exp}[x+2], \{x, -1\}]$

```
FindRoot::lstol: The line search decreased the step size
to within tolerance specified by AccuracyGoal and
PrecisionGoal but was unable to find ...
```

$\{x \to -0.967488\}$

Если построить график функций, стоящих в левой и правой части уравнения, то станет понятно, что решения не существует.

$\mathbf{Plot}[\{x^3, \mathbf{Exp}[x+2]\}, \{x, -4, 4\}, \mathbf{PlotStyle} \to \mathbf{Thickness}[0.01],$
$\mathbf{PlotRange} \to \{\{-4, 4\}, \{-20, 20\}\}]$



Если использовать комплексное число в качестве начального приближения, то функция `FindRoot` вернет комплексный корень (ближайший к начальному приближению).

$\mathbf{FindRoot}[x^3 == \mathbf{Exp}[x+2], \{x, -1 + i\}]$

$\{x \to -1.02483 + 0.930313\, i\}$

Используя функцию `FindRoot` можно находить приближенные решения систем уравнений.

$\mathbf{FindRoot}[\{x^2 + y^2 + x + 2y - 1 == 0, x^2\mathbf{Exp}[x] - y == 0\}, \{x, 0.1\}, \{y, 0.5\}]$

```
{x→0.381748,y→0.213474}
```

Имеется несколько необязательных аргументов (опций) у функции `FindRoot`. Опция `AccuracyGoal->m` определяет точность (количество цифр m) результата. Опция `WorkingPrecision->n` определяет точность (количество цифр n) внутренних вычислений.

$$\mathbf{FindRoot}[x^3 == \mathbf{Exp}[x+2] - 5, \{x, -1\}, \mathbf{AccuracyGoal} \to 24, \mathbf{WorkingPrecision} \to 44]$$

$$\{x \to -1.4953271605616215343618924248534539984950606\}$$

Если окажется, что шагов итерации недостаточно для получения требуемой точности, то их количество можно увеличить с помощью опции `MaxIterations->p`.

$$\mathbf{FindRoot}[x^4 == 0, \{x, 0.023\}, \mathbf{AccuracyGoal} \to 24, \mathbf{WorkingPrecision} \to 34]$$

После выполнения приведенного кода вы получите сообщение, что процесс определения корня остановился после 15 итераций и требуемая точность не достигнута. Тогда увеличьте число итераций следующим образом:

$$\mathbf{FindRoot}[x^4 == 0, \{x, 0.023\}, \mathbf{AccuracyGoal} \to 24, \mathbf{WorkingPrecision} \to 34, \mathbf{MaxIterations} \to 200]$$

$$\{x \to 2.3578175336439518783868878809363 5 \times 10^{-24}\}$$

С другими, реже используемыми функциями решения алгебраических уравнений, вы сможете познакомиться по справочной системе.

1.5 Функции и выражения в системе Mathematica.

В системе имеется огромное количество встроенных функций, однако имеется возможность создавать свои функции. С несколькими простыми способами мы уже познакомились. Простейший способ состоит в выполнении команды

$\boldsymbol{y}[\mathbf{x_}] := $ *выражение*

или

$\boldsymbol{y}[\mathbf{x_}] = $ *выражение* (без двоеточия)

Здесь *y* – имя функции (такое как Sin для функции Sin[x]). Выражение в правой части представляет любой набор операций и вызовов функций, использующих переменную *x* (без символа подчеркивания). Символ подчеркивания в левой части говорит о том, что переменная *x* в правой части выражения является локальной переменной. Знак **:=** представляет отложенное присваивание, которое выполняется в тот момент, когда происходит конкретное вычисление значения функции. Если используется знак **=** , то выполняется создание функции немедленно. Вы должны понимать разницу между термином «функция» и «выражение». Например, Sin – это функция, а Sin[x] – выражение. Аналогично выше мы создали функцию y, а y[x] – это выражение.

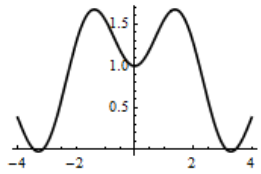
После создания функцию можно использовать. Например, можно вычислять значения функций в точке или строить график.

$$\boldsymbol{y}[\mathbf{x\_}] = \mathbf{Sin}[x]^2 + \frac{\mathbf{Sin}[x]}{x};$$

$$\{\boldsymbol{y}[0.0001], \boldsymbol{N}[\boldsymbol{y}[1]], \boldsymbol{N}[\boldsymbol{y}[\frac{\pi}{2}]]\}$$

$$\{1., 1.54954, 1.63662\}$$

$$\mathbf{Plot}[\boldsymbol{y}[x], \{x, -4, 4\}]$$



Обратите внимание, что при построении графика используется выражение y[x]. Такую функцию можно дифференцировать (на самом деле мы дифференцируем выражение y[x], получаемое из функции).

$\boldsymbol{D[y[x], x]}$

$\frac{\text{Cos}[x]}{x} - \frac{\text{Sin}[x]}{x^2} + 2\text{Cos}[x]\text{Sin}[x]$

При создании функции можно использовать другие функции

$\mathbf{ex[x\_] := Expand[(1 + x)^2]}$

$\mathbf{ex[x]}$

$1 + 2x + x^2$

Операция «;» – точка с запятой объединяет несколько выражений в одно так, что результатом является результат последнего выражения в цепочке. Тогда функцию можно создавать так

$\boldsymbol{y[x\_] = (b = x + x^2; c = x + b; c + b);}$

$\boldsymbol{y[x]}$

$3x + 2x^2$

О любой функции, встроенной или созданной пользователем, системе можно задать вопрос. Для этого надо набрать имя функции, перед которым поставить вопросительный знак.

? **Sin**

Sin[z] gives the sine of z.>>

? **ex**

Global`ex

ex[x_] := Expand[(1 + x)^2]

Более подробную информацию о функции можно получить, если поставить два вопросительных знака перед именем функции.

?? **Sin**

Sin[z] gives the sine of z.>>

Attributes[Sin] = {Listable, NumericFunction, Protected}

?? $\boldsymbol{y}$

Global`y

$y[x\_] = 3x + 2x^2$

Общий способ создания функции пользователя состоит в использовании функции `Function`.

```
funcname = Function[x, body]
```

Эта форма предназначена для определения простой функции с одним формальным параметром (переменной) *x*. Здесь `funcname` имя функции, такое как Sin, Log и т.д., *body* – выражение, определяющее правило вычисления. Например

$\boldsymbol{z = \mathbf{Function}[x, x^2]};$
$\boldsymbol{z[2]}$
4
Здесь z – имя функции, первое вхождение x в правой части – имя формального параметра, затем следует выражение, описывающее правило вычисления значения функции. Выражение может состоять из нескольких выражений, разделенных точкой с запятой. Результат вычисления последнего выражения в такой последовательности будет результатом вычисления функции.
$\mathbf{f1 = \mathbf{Function}[x, g = x^2; g * \mathbf{Sin}[x]]};$
$\mathbf{Plot}[\mathbf{f1}[x], \{x, 0, 4\}]$
С функцией, созданной таким образом, можно работать как с любой другой. Например, такую функцию можно дифференцировать
$\boldsymbol{D}[\mathbf{f1}[x], x]$
$x^2\mathrm{Cos}[x] + 2x\mathrm{Sin}[x]$
При создании можно не определять имя функции.
$\mathbf{Function}[x, x^2];$
Тогда можно использовать знак % вместо имени функции (% представляет результат выполнения последней операции в системе).
$\%[n]$
n^2
Функцию можно определить без указания имен ее аргументов. Тогда для обращения к ним следует использовать значки #1, #2 и т.д.
`funcname=Function[body]`
где `funcname` имя функции, *body* – выражение, определяющее правило вычисления. Например,
$\mathbf{f2 = \mathbf{Function}[\#1^2]}$
$\#1^2\&$
Здесь #1 заменяет имя первого (и единственного) аргумента. Тогда в выражениях можно использовать `f2[x]` или вместо *x* подставлять конкретные значения `f2[2]`.

Синонимом основного определения функции является ее постфиксное определение в виде
$\mathbf{funcname} = (\#1^2)\&$
$\#1^2\&$
Формальные параметры при определении функции обозначаются # (или #1, #2 и т.д.). Само имя функции обозначается #0, ## используется для обозначения всего списка аргументов, ##n обозначает списка аргументов, начиная с n-того. Значок **&** используется вместо ключевого слова `Function` после указания тела функции. Таким образом, последнее определение эквивалентно постфиксному использованию `Function`.

$$\mathbf{funcname} = \left(\#1^2\right)\overbrace{\phantom{//\mathrm{Function}}}^{\&}//\mathrm{Function}$$

Следующие примеры иллюстрируют применение описанных правил и обозначений.

$\mathbf{Function}[x, x^2][5]$ вернет 25.

$(\# + 1)\&[x]$ вернет 1+x.

$F[\#\#, \#\#2]\&[x, y, z]$ даст $F[x, y, z, y, z]$

Вероятно, последняя команда вернет другой результат, поскольку в текущей сессии работы системы *Mathematica* мы переменным y и z присваивали некоторые значения. Чтобы система забыла определение переменных можно, например, выполнить команду

$\mathbf{Remove}[x, y, z, f, F]$

где в список «забываемых» переменных мы на всякий случай включили и другие имена. После этого повторите команду $F[\#\#, \#\#2]\&[x, y, z]$.

Для определения функции с несколькими аргументами можно использовать форму

`Function[{x1, x2,…},body]`

с использованием списка формальных параметров `{x1,x2,…}`. Можно использовать и простую форму, например

$z[x\_, y\_] = x^2 + y^2;$

$D[z[x, y], x]$

$2x$

$\mathbf{Plot3D}[z[x, y], \{x, -1, 1\}, \{y, -1, 1\}]$

$z[2, 3]$

13

Есть еще один способ создания функции с использованием символа отображения (стрелки)

$z = x \to x^3$

$\mathrm{Function}[x, x^3]$

Здесь стрелка особая! Чтобы правильно ее набрать надо последовательно нажать комбинацию клавиш $Esc - fn - Esc$.

Можно создавать рекуррентные функции. Например, функция факториала может быть создана следующим образом

$f = \mathbf{If}[\#1 == 1,\ 1, \#1 * \#0[\#1 - 1]\ ]\&;$

$f[5]$

120

Здесь в определении функции #1 заменяет имя первого (и единственного) аргумента функции, а #0 – используется вместо имени функции `f`. Таким образом, запись $\#0[\#1 - 1]$ означает вызов той же функции `f`, но с аргументом на единицу меньшим.

В *Mathematica* есть несколько способов применить функцию к выражению. Это обычные квадратные скобки f[x], префикс f@x и постфикс x // f. Например,

$\{\mathbf{Sin}\left[\frac{\pi}{6}\right],\ \mathbf{Sin}@\frac{\pi}{4},\ \frac{\pi}{3}//\mathbf{Sin}\}$

$\left\{\frac{1}{2}, \frac{1}{\sqrt{2}}, \frac{\sqrt{3}}{2}\right\}$

Префиксное и постфиксное использование имени функции допустимо для функций одной переменной. Для функции двух переменных возможно инфиксное (между аргументами) использование имени функции. Например

$\boldsymbol{f}[\mathbf{x\_}, \mathbf{y\_}] = \boldsymbol{x} + \boldsymbol{y}^{\mathbf{2}}$

$2 \sim f \sim 3$

11

или

$1 \sim f \sim 2 \sim f \sim 3$

14

Для некоторых функций двух переменных имеются специальные обозначения при использовании в инфиксной форме. Например, в уравнениях используется знак ==, являющийся инфиксной формой функции Equal. Знак → является инфиксной формой функции Rule.

Вам часто придется обращаться к элементам выражений, которые создает система, и потребуется выделять части выражений. К элементам выражений мы можем обращаться как к элементам списков по индексу. Например

$\boldsymbol{z} = \boldsymbol{f}[\boldsymbol{g}[\boldsymbol{a}], \boldsymbol{h}[\boldsymbol{b}]]$

$\{\boldsymbol{z}[[\mathbf{0}]], \boldsymbol{z}[[\mathbf{1}]], \boldsymbol{z}[[\mathbf{2}]], \boldsymbol{z}[[\mathbf{2}, \mathbf{0}]], \boldsymbol{z}[[\mathbf{2}, \mathbf{1}]]\}$

$\{f, g[a], h[b], h, b\}$

Здесь для выделения элементов выражения `z` мы используем индексацию. Так `z[[0]]=f` – имя внешней функции; `z[[1]]=g[a]` – первый аргумент функции, `z[[2]]=h[b]` – второй аргумент; `z[[2,0]]=h` – имя функции, стоящей во втором аргументе; `z[[2,1]]=b` – аргумент второй функции.

Например, создадим функцию

$\boldsymbol{f} = \mathbf{Function}[\{\boldsymbol{x}\}, \boldsymbol{t} = \boldsymbol{x}^{\mathbf{3}}; \boldsymbol{t} + \mathbf{1}];$

Тогда

$\{\boldsymbol{f}[[\mathbf{0}]], \boldsymbol{f}[[\mathbf{1}]], \boldsymbol{f}[[\mathbf{2}]]\}$

$\{\text{Function}, \{x\}, 1 + x^3\}$

Таким образом, обращаясь к имени f , используя индексацию, мы выделяем элементы в правой части выражения. Вот еще один пример.

$\boldsymbol{s} = \mathbf{Factor}[\boldsymbol{x}^{\mathbf{4}} - \mathbf{3}^{\mathbf{4}}]$

$(-3 + x)(3 + x)(9 + x^2)$

$\{\boldsymbol{s}[[\mathbf{1}]], \boldsymbol{s}[[\mathbf{2}]], \boldsymbol{s}[[\mathbf{3}]]\}$

$\{-3 + x, 3 + x, 9 + x^2\}$

или

$\boldsymbol{p} = \mathbf{Factor}[\boldsymbol{x}^{\mathbf{5}} - \mathbf{2}^{\mathbf{5}}]$

$(-2 + x)(16 + 8x + 4x^2 + 2x^3 + x^4)$

$\mathbf{Last}[\boldsymbol{p}]$

$16 + 8x + 4x^2 + 2x^3 + x^4$

Создание функций является ответственным моментом и скорее относится к разделу программирования. Здесь мы дали только введение в способы создания и использования функций.