

***Тема: Моделирование
физических
явлений при помощи
пакета Mathematica***



II. Типы данных

- 1. Основные классы данных**
- 2. Выражения**
- 3. Списки и массивы**
- 4. Объекты и идентификаторы**
- 5. Функции, опции, атрибуты и директивы**
- 6. Подстановки**
- 7. Функции линейной алгебры**

Основные классы данных

Mathematica оперирует с тремя основными классами данных:

- 1. численными данными, представляющими числа различного вида;**
- 2. символьными данными, представляющими символы, тексты и математические выражения (формулы);**
- 3. списками — данными в виде множества однотипных или разнотипных данных.**

Основные классы данных

Численные данные

а) Двоичные числа - 0 или 1.

б) Десятичные числа

Обозначение	Тип чисел	Примеры задания	
Integer	целочисленные	123	-345
Rational	рациональные	123/567	-23/67
Real	вещественные	123.	-123.456 10^6
Complex	комплексные	-3.5 + 0.56i	I

Основные классы данных

г) Данные вещественного типа

Численные данные можно представить также десятичными вещественными числами, которые могут иметь различную форму, например, 123.456, $1.234 \cdot 10^2$, $12345.6 \cdot 10^{-2}$ и т.д. В общем случае они содержат мантиссу с целой и дробной частями и порядок, вводимый как степень числа 10. Выражение с порядком выделяется из мантиссы знаком умножения – либо пробелом, либо звездочкой *. В вводе для возведения в степень используется символ ^, но стандартный вывод задает порядок с применением чисел порядка в отдельной строке над мантиссой. Например:

```
In[1]:= 23.456*10^100
```

```
Out[1]= 1.23456 10102
```

```
In[2]:= N[Sin[1]]
```

```
Out[2]= 0.841471
```

Основные классы данных

Mathematica производит операции с числами изначально как с целыми. Однако установка значка разделительной точки означает, что число должно рассматриваться как вещественное. Например, `1` — целое число, но `1.` — уже вещественное число.

Для представления выражения `expr` в форме вещественного числа используется функция

`N [expr]` или `N [expr, число_цифр_результата]`.

`N[2*Pi,50]`

6.283185307179586476925286766559005768394338

Основные классы данных

Функции `IntegerPart [x]` и `FractionalPart [x]` обеспечивают возврат целой и дробной частей вещественного числа x :

`N[Pi]`

3.14159

`IntegerPart[Pi]`

3

`FractionalPart[Pi]`

-3.+ Л

`N[FractionalPart[Pi]]`

0.141593

Функция `RealDigits [x]` возвращает список реальных цифр результата и число цифр целой части x :

`RealDigits[N[2*Pi]]`

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

Основные классы данных

д) Комплексные числа

Многие математические операции базируются на понятии комплексных чисел, имеющих действительную и мнимую части. Они задаются в форме

$$z = \text{Re}(z) + I * \text{Im}(z) \text{ или } z = \text{Re}(z) + i \text{Im}(z)$$

где знак I - мнимая единица (квадратный корень из -1), $\text{Re}(z)$ - действительная часть комплексного числа, а $\text{Im}(z)$ - мнимая часть.

$$2 + I * 3 \text{ или } 2 + 3I$$

Мнимая часть задается умножением ее значения на символ мнимой единицы I – (или i). При этом знак умножения $*$ можно указывать явно или заменить пробелом. Части комплексного числа могут быть целыми, вещественными и рациональными числами.

Основные классы данных

е) Символьные данные и строки

Символьные данные в общем случае могут быть отдельными символами (пример a, b,...,z), строками (strings) и математическими выражениями expr (от expression - выражение), представленными в символьном виде. Символьные строки задаются цепочкой символов в кавычках, например, 0 "ssss", и являются строчными объектами.

`\n`— новая строка (line feed);

`\t` — табуляция.

Примеры:

"Hello my friend!"

Hello my friend!

"Hello\nmy\nfriend!"

Hello

my

friend!

"Hello\tmy\tfriend!"

Hello my friend;

Основные классы данных

ж) Выражения

Выражения в системе Mathematica обычно ассоциируются с математическими формулами, например:

Запись на языке Mathematica	Обычная математическая запись
<code>2*Sin[x]</code>	$2 \cdot \sin(x)$
<code>2 Sin[x]</code>	$2 \cdot \sin(x)$
<code>(a + b^2 + c^3) / (3*d - 4*e)</code>	$(a + b^2 + c^3) / (3d - 4e)$
<code>sqrt(2)</code>	$\sqrt{2}$
<code>Integrate [Sin [x] , x]</code>	$\int \sin(x) dx$

Основные классы данных

Для записи математических выражений используются как операторы, так и функции. Тонкости синтаксиса системы, используемого при записи арифметических операций следующие:

1. знак умножения может быть заменен пробелом;
2. встроенные функции начинаются с прописной буквы и обычно повторяют свое общепринятое математическое обозначение;
3. круглые скобки () используются для выделения частей выражений и задания приоритета при их выполнении;
4. параметры функций задаются в квадратных скобках [];
5. фигурные скобки {} используются при задании списков.

Основные классы данных

3) Списки и массивы

Наиболее общим видом сложных данных в системе являются списки (иногда их называют листами) *Списки* представляют собой совокупность однотипных или разнотипных данных, сгруппированных с помощью фигурных скобок.

$\{1,2,3\}$	список из трех целых чисел;
$\{a,b,c\}$	список из трех символьных данных;
$\{1,a,x^2\}$	список из разнотипных данных;
$\{\{a,b\},\{c,d\}\}$	список, эквивалентный матрице;
$\{x^2+y^2,2x \sin[x]\}$	список из двух математических выражений.

Основные классы данных

Объекты и идентификаторы

В общем случае система Mathematica оперирует с объектами. Под ними подразумеваются математические выражения (`expr`), символы (`symbols`), строки из символов (`strings`), упомянутые выше числа различного типа, константы, переменные, графические и звуковые объекты и т. д.

Каждый объект характеризуется своим именем — идентификатором. Это имя должно быть уникальным, то есть единственным. Существуют следующие правила задания имен:

- ✓ `sssss` — имя объекта, заданного пользователем;
- ✓ `Sssss` — имя объекта, входящего в ядро системы;
- ✓ `$Sssss` — имя системного объекта.

Основные классы данных

Константы

Константы являются типовыми объектами системы, несущими заранее определенное численное или символьное значение. Это значение не должно меняться по ходу выполнения документа. К численным константам относятся любые числа, непосредственно используемые в математических выражениях или в программных объектах, например, в процедурах и функциях. Используются следующие поименованные константы:

ComplexInfinity	комплексная бесконечность, которая представляет величину с бесконечным модулем и неопределенной фазой;
Degree	число радиан в одном градусе, которое имеет числовое значение $\pi/180$;
E	основание натурального логарифма с приближенным числовым значением 2.71828;
Infinity	“положительная” бесконечность (со знаком минус – “отрицательная” бесконечность;
Pi	- число $\pi=3,14159$
и другие.	

Основные классы данных

Переменные

Переменные - это объекты, которые могут принимать различные значения, находящиеся в определенном допустимом множестве значений.

Переменным обычно присваиваются имена- идентификаторы.

var=value

var - имя переменной; **value** - ее значение.

Основные операции по присвоению переменным их значений в упрощенной форме:

x=value	- переменной x присваивается значение value, и оно выводится в строку вывода;
x=y=...=value	- значение value присваивается переменным x,y... и выводится в строку вывода;
x:=value	- значение value присваивается переменной x в невычисленной форме и не выводится в строку вывода;

Основные классы данных

Примеры по присвоению значений переменной:

$g = \text{Plot}[\text{Sin}[x], \{x, 0, 20\}]$	– переменной g присваивается значение в виде графического объекта;
$y = 1 + x^2$	– переменной y присваивается символьное значение в виде математического выражения $(1 + x^2)$;
$z = \{1, 2, x, a + b\}$	– переменной z присваивается значение в виде списка, содержащего четыре элемента.

Основные классы данных

Функции системы

Функцией в системе Mathematica называют объект, который задается своим именем и списком формальных параметров в квадратных скобках и возвращает в ответ на обращение к нему по имени с указанием списка фактических параметров, некоторый результат. Он может иметь тип числа, строчного выражения или математического выражения - формулы.

Под это определение попадает множество хорошо известных функций, например, алгебраические $\ln(x)$ или $\exp(x)$, тригонометрические $\sin(x)$, $\cos(x)$ и т.д. На языке Mathematica они записываются как `Ln[x]`, `Exp[x]`, `Sin[x]`, `Cos[x]` и т.д.

В общем виде функцией является объект вида:

Идентификатор_функции[o1,o2,o3,...]

Здесь `o1`, `o2`, `o3`,... - объекты-параметры, которые могут быть числами, константами, списками, математическими выражениями, символами, строками, математическими выражениями, символами, строками, опциями и т.д. (Таблица 1).

Основные классы данных

Таблица 1. Математические функции в системе Mathematica

Запись в математике	Реализуемое действие	Запись в системе Mathematica
$ x $	абсолютное значение	Abs[x]
$\cos x$	косинус X	Cos[x]
$\sin x$	синус X	Sin[x]
$\operatorname{tg} x$	тангенс X	Tan[x]
$\operatorname{ctg} x$	котангенс X	Got[x]
e^x	экспонента X	Exp[x]
$\ln x$	логарифм натуральный	Log[x]
$\log x$	логарифм десятичный	Log[10,x]
\sqrt{x}	корень квадратный	Sqrt[x]
π	пи	
x^N	X в степени N	

```
In[1]:= Fun[x_]:=Exp[x]-1  
N[Fun[1.0],10]  
Out[1]= 1.718281828
```

Основные классы данных

Функции генерации случайных чисел

Для реализации статистических методов моделирования используются случайные числа. Система имеет генератор псевдослучайных чисел, доступ к которому обеспечивают следующие функции:

`Random []` — возвращает равномерно равномерно распределенное псевдослучайное число типа `Real` в интервале от 0 до 1;

`Random [type, range]` — дает псевдослучайное число указанного типа `type`, лежащее в указанном интервале `range`. К возможным типам относятся `Integer`, `Real` и `Complex`. По умолчанию принят интервал от 0 до 1. Можно задать интервал явно в виде `{min, max}`; спецификация интервала в виде `max` эквивалентна `{0, max}`;

`SeedRandom[n]` — сбрасывает (устанавливает в начальное состояние) генератор случайных чисел, используя целое `n` как начальное число;

`SeedRandom []` — устанавливает генератор, используя в качестве начального числа текущее время.

Основные классы данных

Хотя генерируемые числа не являются строго случайными, их количество в повторяющейся

послед

специа

наприм

послед

Для

массив

случай

соотве

The screenshot shows a Mathematica notebook window with the following content:

Генерация массива из 1000 случайных чисел

```
In[55]:= pts = Table[Point[{Random[], Random[]}], {10000}];
```

```
In[58]:= Show[Graphics[{PointSize[0.008], pts}]]
```

Out[56]= - Graphics -

100%

льзование

ератора,

вторение

БОЛЬШОГО

ПОМОЩЬЮ

ТОЧКИ,

Основные классы данных

Выполнение арифметических операций

Для выполнения простых арифметических операций достаточно набрать необходимое математическое выражение и нажать Enter (справа)

а) арифметические вычисления

```
In[1]:= N[Sin[1]]  
Out[1]= 0.841471
```

```
In[2]:= 2+2  
Out[2]= 4
```

б) использование результатов предыдущих вычислений (%)

% - возвращает результат последней предшествующей операции

%% - возвращает результат операций выполненной перед последней предшествующей операцией.

```
In[3]:= %+4  
Out[3]= 8
```

Основные классы данных

в) вывод результатов вычислений

Большинство функций системы возвращают результат в символьном виде.

```
In[4]:= Sin[1]
```

```
Out[4]= Sin[1]
```

Функция с однобуквенным названием $N[expr]$ дает численное вещественное значение (аппроксимацию) для любого выражения $expr$ (expression - выражение), которое имеет численное значение. $N[\text{Sin}[1]]$

Также можно задать вычисление любого выражения с выводом результатов в форме вещественного числа, используя **expr//**

```
In[5]:= 1/3+2/7
```

```
Out[5]=  $\frac{21}{21}$ 
```

```
In[6]:= 1/3+2/7//N
```

```
Out[6]= 0.619048
```

Основные классы данных

г) арифметические вычисления с повышенной точностью `Rationalize[x]` и `Rationalize[x,dx]` дают приближение для числа x в виде рациональных чисел. Вторая из этих функций задает приближение с заданной точностью.

```
In[7]:= Rationalize[N[Pi],10^-3]
```

355

```
Out[7]=  $\frac{355}{113}$ 
```

Работа со списками и массивами

Списки и их свойства

Часто математические или иные объекты содержат множество данных, которые желательно объединять под заданным именем.

Например, под объектом с именем M можно подразумевать квадратную матрицу 10×10 элементов с их общим числом 100.

Для объединения данных используются списки или листы (список - по английски list).

Mathematica имеет обширные возможности работы со списками, содержащими не только однотипные, но и разнотипные данные – элементы. На языке системы список – это совокупность данных, указанных в фигурных скобках, например:

$\{1,4,2,7,9\}$ или $\{a,b,c,d,e\}$

Списки можно составлять напрямую, задавая объекты в соответствии с описанным синтаксисом. Однако можно и генерировать некоторые виды списков, таких , как таблицы или массивы.

Работа со списками и массивами

Генерация списков

Для генерации списков с элементами - вещественными и целыми числам или даже целыми выражениями - особенно часто используется функция Table, создающая таблицу - список.

Table[expr,{imax}]	- генерирует список, содержащий imax экземпляров выражения expr;
Table[expr,{imax}]	- генерирует список значений expr при i, изменяющимся от 1 до imax с шагом 1;
Table[expr,{imin, imax }]	- генерирует список значений expr при i, изменяющимся от значения i=imin до i=imax с шагом 1;
Table[expr,{imin, imax,di }]	- генерирует список значений expr при i, изменяющимся от значения i=imin до i=imax с шагом di.

Работа со списками и массивами

Примеры на использование функции Table:

Пример	Результат	Комментарий
<code>Table[Exp[i], {5}]</code>	$\{E^i, E^i, E^i, E^i, E^i\}$	Генерация пяти значений E^i ($i=\text{const}$)
<code>Table[i*j, {i, 1, 3}, {j, 1, 3}]</code>	$\{\{1, 2, 3\}, \{2, 4, 6\}, \{3, 6, 9\}\}$	Генерация матрицы размером 3×3

Работа со списками и массивами

Применяется также функция **Range** для создания ранжированных числовых элементов, значения которых лежат в некотором диапазоне числовых значений.

Range[imax]	- генерирует список целых чисел от 1 до imax.
Range[imin,imax]	- генерирует список целых чисел от imin до imax.
Range[imin,imax,di]	- генерирует список вещественных чисел от imin до imax с шагом di.

Примеры на использование функции Range:

Пример	Результат	Комментарий
Range[5]	{1,2,3,4,5}	Генерация пяти целых чисел
Range[0,2,0.5]	{0,0.5,1,1.5,2}	Генерация пяти вещественных чисел

Работа со списками и массивами

Выделение и вывод элементов списков.

Списки представляли бы малую ценность, если бы не было средств выделения любых элементов из них. Но такие средства есть. Для выделения элементов списка используются двойные квадратные скобки.

<code>list[[i]]</code>	- выделяет <i>i</i> -й элемент списка;
<code>list[{{i,j,...}}]</code>	-выделяет <i>i</i> -й, <i>j</i> -й и т.д. элементы списка.

Работа со списками и массивами

Для вывода элементов списка используется функция `TableForm` с рядом опций:

TableForm[list]	- выполняет вывод элементов списка <code>list</code> в табличной форме.
TableAlignments	- опция для <code>TableForm</code> и <code>MatrixForm</code> , которая устанавливает, как должно выравниваться содержимое списка в каждом измерении;
TableDepth	- опция для <code>TableForm</code> и <code>MatrixForm</code> , которая устанавливает максимальное число уровней, выводимых в табличном или матричном формате;
TableDirection	- опция для <code>TableForm</code> и <code>MatrixForm</code> , которая указывает, располагать ли последовательные измерения в виде строк или столбцов;
TableHeadings	- опция для <code>TableForm</code> и <code>MatrixForm</code> , которая дает метки (labels) для каждого измерения таблицы или матрицы;
TableSpacing	- опция для <code>TableForm</code> и <code>MatrixForm</code> , которая устанавливает количество пробелов, которые следует оставлять между соседними строками или столбцами.

Работа со списками и массивами

Эти же опции используются и для функции **MatrixForm[list]**, служащей для вывода матриц. Эта функция печатает элементы списка list следующим образом: каждый элемент списка заключен в квадратную ячейку одинакового с другими размера.

Одноуровневый список печатается в виде столбца, а двухуровневый - в стандартной матричной форме.

Векторы и матрицы являются разновидностью списков, причем векторы – это одномерные массивы, а матрицы – двумерные.

Пример	Результат	Комментарий
<code>l1:={1,2,3,4,5}</code>	<code>{1,2,3,4,5,}</code>	Задание исходного списка l1
<code>l1[[3]]</code>	3	Выделение третьего элемента 3
<code>l1[{{1,2,5}}]</code>	<code>{1,2,5}</code>	Выделение первого, второго и пятого элементов
<code>l2={{1,2,3},{4,5,6}}</code>	<code>{{1,2,3},{4,5,6}}</code>	Задание сдвоенного (двумерного) списка
<code>TableForm[l2]</code>	$\begin{array}{ccc} 1 & 2 & 3 \\ 4 & 5 & 6 \end{array}$	Вывод сдвоенного списка в табличной форме

Работа со списками и массивами

Выявление структуры списков

Списки относятся к данным сложной структуры. Поэтому при работе с ними возникает необходимость контроля за структурой списков, без чего их применение может привести к грубым ошибкам – как явным, сопровождаемым выдачей сообщения об ошибке, так и неявным.

Для выявления структуры списков используется ряд функций:

Lenght[list]	- возвращает число элементов списка list;
Dimensions[list]	- возвращает размерность списка;
MatrixQ[list]	- проверяет, является ли список матрицей, и дает True, если это так, и False в противном случае;
TensorRank[list]	- возвращает ранг списка;
MemberQ[list,form]	- проверяет, есть ли form в списке, и возвращает True, если это так, и False в противном случае;
Count[list,form]	- возвращает число вхождений form в список;
FreeQ[list,form]	- проверяет, свободен ли список от form: если да, то возвращает True, иначе возвращает False;
Position[list,form]	- возвращает номер позиции form в списке;
VectorQ[list]	- проверяет, является ли список вектором, и возвращает True, если это так, и False в противном случае.

Работа со списками и массивами

Примеры на использование этих функций:

Ввод (<i>In</i>)	Вывод (<i>Out</i>)
$I1=\{1,2,3,4,1\}$	
<code>Lenght[I1]</code>	5
<code>Dimensions[I1]</code>	{5}
<code>MatrixQ[I1]</code>	False
<code>TensorRank[I1]</code>	1
<code>MemberQ[I1,1]</code>	True
<code>Count[I1,1]</code>	2
<code>FreeQ[I1,5]</code>	True
<code>Position[I1,1]</code>	{{1},{5}}
<code>VectorQ[I1]</code>	True

Система оставляет за пользователем свободу действий в зависимости от результатов анализа структуры списков.

Работа со списками и массивами

Работа со списком в стеке

Элементы списков могут размещаться в так называемом стеке.

Стек – это особая структура хранения данных, чисто умозрительно напоминающая стопку тарелок в шкафу. В нашем примере тарелки – это данные. Очередную «тарелку» можно положить только сверху – на вершину стека. На дне стека лежит первая введенная в него «тарелка». Стек подчиняется правилу: последнее введенное данное извлекается первым, а первое введенное данное извлекается последним. Стек относится к системам хранения данных динамического типа – его размеры непрерывно меняются по ходу вычислений. Стек может быть и пустым, если из него были извлечены все данные.

Работа со списками и массивами

Работа со стеком

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

Drop[list,n]	- возвращает список list, из которого удалены первые n элементов;
Drop[list, -n]	- возвращает список list, из которого удалены последние n элементов;
Drop[list, {n}]	- возвращает список list без n-го элемента;
Drop[list, {m,n}]	- возвращает список list, отбросив элементы от m до n;
\$ Fold[f,x,list]	- возвращает последний элемент из FoldList[f,x,list];
\$ Last[list]	- возвращает последний элемент списка list;
Rest[list]	- возвращает список с уничтоженным первым элементом;
Take[list,n]	- возвращает первые n элементов списка list;
Take[list,-n]	- возвращает последние n элементов списка list;
Take[list,{m,n}]	- возвращает элементы списка с порядковыми номерами от m до n.

Работа со списками и массивами

Пояснить работу со стеком могут следующие примеры:

Ввод (<i>In</i>)	Вывод (<i>Out</i>)
<code>Drop[{1,2,3,4,5},2]</code>	<code>{3,4,5}</code>
<code>Drop[{1,2,3,4,5},-2]</code>	<code>{1,2,3}</code>
<code>Drop[{a,b,c,d,e},{2,4}]</code>	<code>{a,e}</code>
<code>Last[{1,2,3,4,5}]</code>	<code>5</code>
<code>Rest[{1,2,3,4,5}]</code>	<code>{2,3,4,5}</code>
<code>Take[{1,2,3,4,5},2]</code>	<code>{1,2}</code>
<code>Take[{1,2,a,b,c},-2]</code>	<code>{b,c}</code>
<code>Take[{1,2,3,4,5},{2,4}]</code>	<code>{2,3,4}</code>

Работа со списками и массивами

Включение в список новых элементов

Mathematica дает ряд расширенных возможностей для работы со списками. Для расширения списка путем включения в него новых элементов используются следующие функции:

\$ Append[list,element]	- добавляет элемент в конец списка;
Prepend[list,element]	- добавляет элемент в начало списка;
Insert[list,element,n]	- вставляет элемент в позицию n (отчет позиции ведется с начала листа, а если задано -n, то с конца).

Примеры, которые иллюстрируют применение этих функций:

Ввод (In)	Выход (Out)
$l=\{1,2,3\}$	$\{1,2,3\}$
<code>Append[l,e]</code>	$\{1,2,3,e\}$
<code>Prepend[l,e]</code>	$\{e,1,2,3\}$
<code>Insert[l,e,2]</code>	$\{1,e,2,3\}$

Работа со списками и массивами

Изменение порядка расположения элементов в списке

Помимо добавления в список новых данных имеется возможность изменения порядка расположения данных в списке. Она реализуется следующими операциями:

Flatten[list] \$	- выравнивает (превращает в одноуровневый) список по всем его уровням;
Flatten[list,n] \$	- выравнивает список по его n уровням;
Flatten[list,n,h] \$	- выравнивает список с заголовком h по его n уровням;
Sort[list]	- сортирует элементы списка в каноническом порядке;
Sort[list,p]	- сортирует элементы списка list упорядочения p;
\$ Reverse[list]	- возвращает список с обратным порядком расположения элементов;
\$ RotateLeft[list]	- возвращает список после однократного поворота влево;
\$ RotateLeft[list,n]	- возвращает список после n-кратного поворота влево;
\$ RotateRight[list]	- возвращает список после однократного поворота вправо;
\$ RotateRight[list,n]	- возвращает список после n-кратного поворота вправо;
\$ Transpose[list]	- осуществляет транспозицию (смену строк и столбцов) для двумерного списка;
\$ Transpose[list,n]	- осуществляет транспозицию n-мерного списка;
Union[list1,list2,...]	- возвращает отсортированный список всех различающихся между собой элементов, принадлежащих любому из данных списков list _i ;
Union[list]	- возвращает отсортированный вариант списка list, в котором опущены все дублируемые элементы

Работа со списками и массивами

Примеры на использование этих функций:

Ввод (In)	Вывод (Out)
<code>l3={{1,2,3},{4,5,6},{7,8,9}}</code>	
<code>Flatten[l3]</code>	
<code>Sort[list]</code>	<code>{e,1,2,3}</code>
<code>Sort[{1,5,3,4,2}]</code>	<code>{1,e,2,3}</code>
<code>Reverse[{1,2,3,4}]</code>	<code>{4,3,2,1}</code>
<code>RotateLeft[{1,2,3,4,5},2]</code>	<code>{3,4,5,1,2}</code>
<code>RotateRight[{1,2,3,4,5},2]</code>	<code>{4,5,1,2,3}</code>
<code>l2={{a,b},{c,d}};</code>	
<code>Union[list1,list2,...]</code>	<code>{1,2,3,4,5,7}</code>
<code>Union[list]</code>	<code>{1,2,3,4}</code>

Работа со списками и массивами

Иногда возникает необходимость комбинирования нескольких списков. Для этого используются операции:

\$ Intersection[list1,list2,...]	- возвращает упорядоченный список элементов, общих для всех списков;
Join[list1,list2,...]	- объединяет списки в единую цепочку, может применяться на любом множестве выражений, имеющих один заголовок.

Приведем примеры:

Ввод (In)	Вывод (Out)
<code>l1={1,2,3,4,5}; l2={a,b,3,4,c};</code>	
<code>Intersection[l1,l2]</code>	<code>{3,4}</code>
<code>Join[l1,l2]</code>	<code>{1,2,3,4,5,a,b,3,4,c}</code>

Работа со списками и массивами

Для работы со списками используются также следующие, менее распространенные функции.

Cases [{e1,e2,...},pattern]	- возвращает список тех e _i , которые соответствуют заданному шаблону (pattern);
\$ Complement [eall,e1,e2,...]	- возвращает элементы eall, которых нет среди e _i ;
ComposeList [{f1,f2,...},x]	- формирует список формы {x,f1[x],f2[f1[x]],...};
Count [list,pattern]	- возвращает количество элементов в списке list, соответствующих образцу pattern;
\$ DeleteCases [{e1,e2,...}, pattern]	- удаляет из списка элементы, которые соответствуют заданному шаблону (pattern);
\$ FoldList [f,x,{a,b,...}]	- возвращает {x,f[x,a],f[f[x,a],b],...};
Position [expr,pattern]	- возвращает список позиций в expr, в которых размещаются объекты, сопоставимые с указанным шаблоном pattern;
Select [list,crit]	- выбирает все элементы e _i списка list, для которых crit[e _i] имеет значение True;
Select [list,crit,n]	- выбирает из первых n элементов, для которых crit[e _i] есть True.

Работа со списками и массивами

Массивы - списки

Совокупность данных образует массив (Array). Массивы могут быть одномерными (один список), двумерными и многомерными (два списка и более). Одномерные массивы в математике называют векторами, двумерные - матрицами. Mathematica позволяет создавать многомерные массивы - число элементов в них ограничено лишь объемом памяти компьютера.

Для задания массивов используются следующие функции:

Array[f,n]	- генерирует список длиной n с элементами $f[i]$, где i имеет значения от 1 до n.
Array[f,{n1,n2,...}]	- генерирует массив с размерностью $n1*n2* \dots$ в виде вложенных списков с элементами $f[i1,i2,...]$;
Array[f,dims,origin]	- генерирует список с размерностью $dims$, используя спецификацию индекса $origin$;
Array[f,dims,origin,h]	- использует заголовок h , а не List, для каждого уровня массива.

Работа со списками и массивами

Примеры задания массивов и их вывода:

Ввод (<i>In</i>)	Вывод (<i>Out</i>)
$Y := \text{Array}[\text{Exp}, 4]$	
Y	$\{E^2, E^3, E^4, E\}$
$N[Y]$	$\{2.71828, 7.38906, 20.0855, 54.5982\}$
$\text{Array}[\text{Sin}, 3, 0]$	$\{0, \text{Sin}[1], \text{Sin}[2]\}$
$\text{Array}[f, 5, 2, 2]$	$2[f[2], f[3], f[4], f[5], f[6]]$

Работа со списками и массивами

Функции для операции с массивами

Основные операции над массивами и матрицами позволяет осуществить следующая группа функций:

Det[m]	- возвращает детерминант (определитель квадратной матрицы m ;
DiagonalMatrix[list]	- возвращает диагональную матрицу с главной диагональю, сформированной из элементов списка list , и нулевыми остальными элементами матрицы;
Dot[a,b,c]	- возвращает произведения векторов, матриц и тензоров (эту операцию можно задавать также и в виде a.b.c);
Eigensystem[m]	- возвращает список {values,vectors} собственных значений и принадлежащих им собственных векторов для квадратной матрицы m ;
Eigenvalues[m]	- возвращает список собственных значений квадратной матрицы m ;
Eigenvectors[m]	- возвращает список собственных векторов квадратной матрицы m ;
IdentityMatrix[n]	- возвращает единичную матрицу с размером $n \times n$ (ее диагональные элементы имеют значения 1, остальные 0);
Inverse[m]	- возвращает обратную матрицу для квадратной матрицы m , т.е. матрицу m^{-1} . Будучи умноженной на исходную матрицу m , эта матрица дает единичную матрицу;
LinearSolve[m,b]	- возвращает вектор x - решение матричного уравнения $m.x==b$, где m – матрица коэффициентов левой части системы линейных уравнений, x – вектор неизвестных и b – вектор свободных членов в правой части системы;
MatrixExp[m]	- возвращает экспоненциал матрицы m ;
MatrixPower[m,n]	- возвращает n -ю степень матрицы m ;
и другие.	

Работа со списками и массивами

Примеры применения основных из этих функций:

Ввод (<i>In</i>)	Вывод (<i>Out</i>)
A:=IdentityMatrix[3]	
A	{{1,0,0},{0,1,0},{0,0,1}}
MatrixExp[A]	{{E,0,0},{0,E,0},{0,0,E}}
MatrixPower [MatrixExp{A},-1.5]	{{0.22313,0,0},{0,0.22313,0},{0,0,0.22313}}
m:={{1,2},{3,7}}	
MatrixForm[m]	$\begin{matrix} 1 & 2 \\ 3 & 7 \end{matrix}$
Det[m]	1
Eigensystem[m]	{{4-Sqrt[15],4+Sqrt[15]}, {{ $\frac{-2}{-3+\text{Sqrt}[15]}$,1},{ $\frac{-2}{-3-\text{Sqrt}[15]}$,1}}}
Inverse[m]	{{7,-2},{-3,1}}

Расширенные математические возможности

Суммы и произведения

Mathematica обладает широкими возможностями в реализации расчетов, относящихся к различным разделам математического анализа. В частности, система позволяет вычислять суммы и произведения вида:

$$\sum_{i=i \min}^{i \max} f_i$$

$$\prod_{i=i \min}^{i \max} f_i$$

В этих операциях суммируются или перемножаются значения некоторой функции f_i . Управляющая переменная i в общем случае принимает значения от минимального (начального) $i \min$ до максимального (конечного) $i \max$ с шагом равным di . При целочисленных значениях i эта переменная называется индексной и прямо указывает на индекс соответствующего элемента ряда.

Расширенные математические возможности

Вычисление сумм

Для вычисления сумм в системе предусмотрен ряд функций:

Sum[f,{i,imax}]	- вычисляет сумму значений f при изменении индекса i от 1 до $imax$ с шагом +1.
Sum[f,{i,imin,imax}]	- вычисляет сумму значений f при изменении индекса i от минимального значения $i=imin$ до максимального $i=imax$ с шагом +1.
Sum[f,{i,imin,imax,di}]	- вычисляет сумму значений f при изменении управляющей переменной вещественного типа от минимального значения $i=imin$ до максимального $i=imax$ с шагом di .
Sum[f,{i,imin,imax},{j,jmin,jmax}]	- вычисляет многократную сумму значений f при изменении индексов i от $imin$ до $imax$ с шагом +1, j от $jmin$ до $jmax$ с шагом +1 (число индексных переменных не ограничено).
NSum[f,{i,imin,imax}]	- возвращает численное значение суммы f при индексе i , изменяющемся от $imin$ до $imax$ с шагом +1.
Nsum[f,{i,imin,imax,di}]	- возвращает сумму численных значений суммы f при индексе i , изменяющемся от $imin$ до $imax$ с шагом di .

Расширенные математические возможности

Приведем примеры использования основных функций суммирования:

Ввод (<i>In</i>)	Вывод (<i>Out</i>)
$\text{Sum}[i^2, \{i, 10\}]$	385
$\text{Sum}[i^2, \{i, 0, 1, 0.25\}]$	1.875
$\text{Sum}\{i^*j, \{i, 1, 10\}, \{j, 2, 5\}\}$	249568
$\text{NSum}[i^2, \{i, 5\})$	55

Таким образом, эти функции обеспечивают расширенные возможности вычисления сумм - как при целочисленных, так и вещественных значениях управляющих переменных переменных, задающих циклы вычислений.

Расширенные математические ВОЗМОЖНОСТИ

Вычисление произведений

Операции вычисления произведений представлены следующими функциями:

Product[f,{i,imax}]	- численное значение произведений значений f при изменении индекса i от 1 до $imax$ с шагом +1.
Product [f,{i,imin,imax}]	- возвращает численное значение произведения f при изменении i от $imin$ до $imax$ с шагом +1.
Product [f,{i,imin,imax,di}]	- возвращает численное значение произведения f при i , изменяющемся от $imin$ до $imax$ с шагом di .
Product[f,{i,imin,imax}, {j,jmin,jmax}]	- вычисляет многократное произведение по нескольким переменным.

Расширенные математические ВОЗМОЖНОСТИ

Примеры использования функций вычисления произведения:

Ввод (<i>In</i>)	Вывод (<i>Out</i>)
Product [i,{i,10}]	3628800
NProduct [k^2,{k,1,5}]	14400.
NProduct [i^2,{1,1,2,0.2}]	93.6405
Product [Log[i],{i,2,5,0.5}]	4.23201 Log[2]

Расширенные математические ВОЗМОЖНОСТИ

Вычисление производных

К числу наиболее часто используемых математических операций принадлежит вычисление производных функций $f(x)$

$$f'(x) = \frac{df(x)}{dx}$$

как в аналитической, так и в символьной форме. Для этого используются следующие функции:

Расширенные математические ВОЗМОЖНОСТИ

D[f,x]	- возвращает частную производную функции f по переменной x ;
D[f,{x,n}]	- возвращает частную производную n -го порядка по переменной x ;
D[f,x1,x2,...]	- возвращает смешанную производную;
Derivative[n1,n2,...][f]	- обобщенная форма дифференцирования функции f , получаемая в результате дифференцирования f $n1$ раз по первому аргументу, $n2$ раза по второму аргументу и т.д.
Dt[f,x]	- возвращает обобщенную производную функции f по переменной x ;
Dt[f]	- возвращает n -ю обобщенную производную по переменной x .
Dt[f]	- возвращает n -ю обобщенную производную по переменной x .
Dt[f,x1,x2,...]	- возвращает смешанную обобщенную производную.

Расширенные математические ВОЗМОЖНОСТИ

Примеры вычисления производных:

Ввод (<i>In</i>)	Вывод (<i>Out</i>)
$D[(x+1)^*(x^2+2),x]$	$2+x^2+2x(1+x)$
$D[x^a,x,a]$	$x^{-1+a}+ax^{-1+a}\text{Log}[x]$
$Dt[x^4,x,x]$	$12x^2$

Расширенные математические ВОЗМОЖНОСТИ

Вычисление первообразных и определенных интегралов

Для вычисления интегралов в системе используются следующие функции:

Integrate[f,x]	- возвращает первообразную (неопределен-ный интеграл) подынтегральной функции f по переменной x ;
Integrate[f,{x,xmin,xmax}]	- возвращает значение определенного интеграла с пределами $a=xmin$ до $b=xmax$;
Integrate[f,{x,xmin,xmax}, {y,ymin,ymax},...]	- возвращает значение кратного интеграла с пределами от $xmin$ до $xmax$ по переменной x , от $ymin$ до $ymax$ по переменной y (кратность реально не ограничена).

Расширенные математические ВОЗМОЖНОСТИ

Для обозначения реальных пределов используется константа Infinity, означающая положительную бесконечность. Пределы могут задаваться как константами так и функциями.

Ввод (In)	Вывод (Out)
<code>Integrate[a*x^b, x]</code>	$\frac{ax^{1+b}}{1+b}$
<code>Integrate[x*sqrt[1+x], {x,0,3}]</code>	$\frac{116}{15}$
<code>Integrate[1/(x*sqrt[x^2-1]), {x,1,Infinity}]</code>	$\frac{\text{Pi}}{2}$

Расширенные математические ВОЗМОЖНОСТИ

Для вычисления численных значений определенных интегралов используется функция

NIntegrate[f,{x,xmin,xmax}]

Она возвращает приближенное значение интеграла от функции f по переменной x в интервале от x_{\min} до x_{\max} .

NIntegrate[, {x, 0.5, 2}]

Расширенные математические

Untitled-4 *

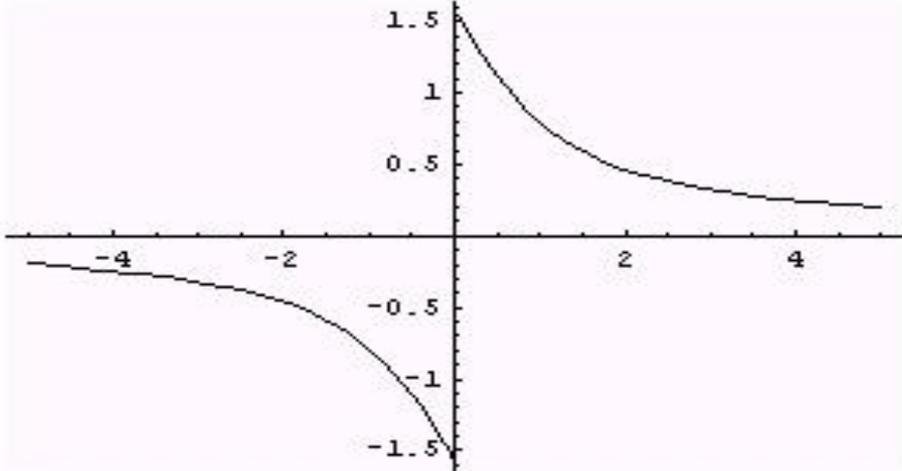
```
In[12]:= Limit[ArcTan[1/x], x -> 0, Direction -> +1]
```

Out[12]= $-\frac{\pi}{2}$

```
In[14]:= Limit[ArcTan[1/x], x -> 0, Direction -> -1]
```

Out[14]= $\frac{\pi}{2}$

```
In[15]:= Plot[ArcTan[1/x], {x, -5, 5}]
```



Out[15]= - Graphics -

100%

функций,
предел в

к 0.

ut)

Вычисл

Сист
заданны
аналитич

Она еще

Вв

Lin

Lin

Lin

Расширенные математические возможности

При работе с функцией `Limit` используются следующие опции:

Direction	указывает направление, в котором возможно приближение к пределу. Опция используется в виде <code>Direction ->-1</code> (или <code>+1</code>);
Automatic	задает автоматический выбор вида функции при вычислении предела.

Применение этих опций может оказаться полезным в сложных случаях вычисления пределов.

Расширенные математические возможности

Решение

Многие

вида

Эти уравнения

Разумеется,

решения уравнения

В этом случае

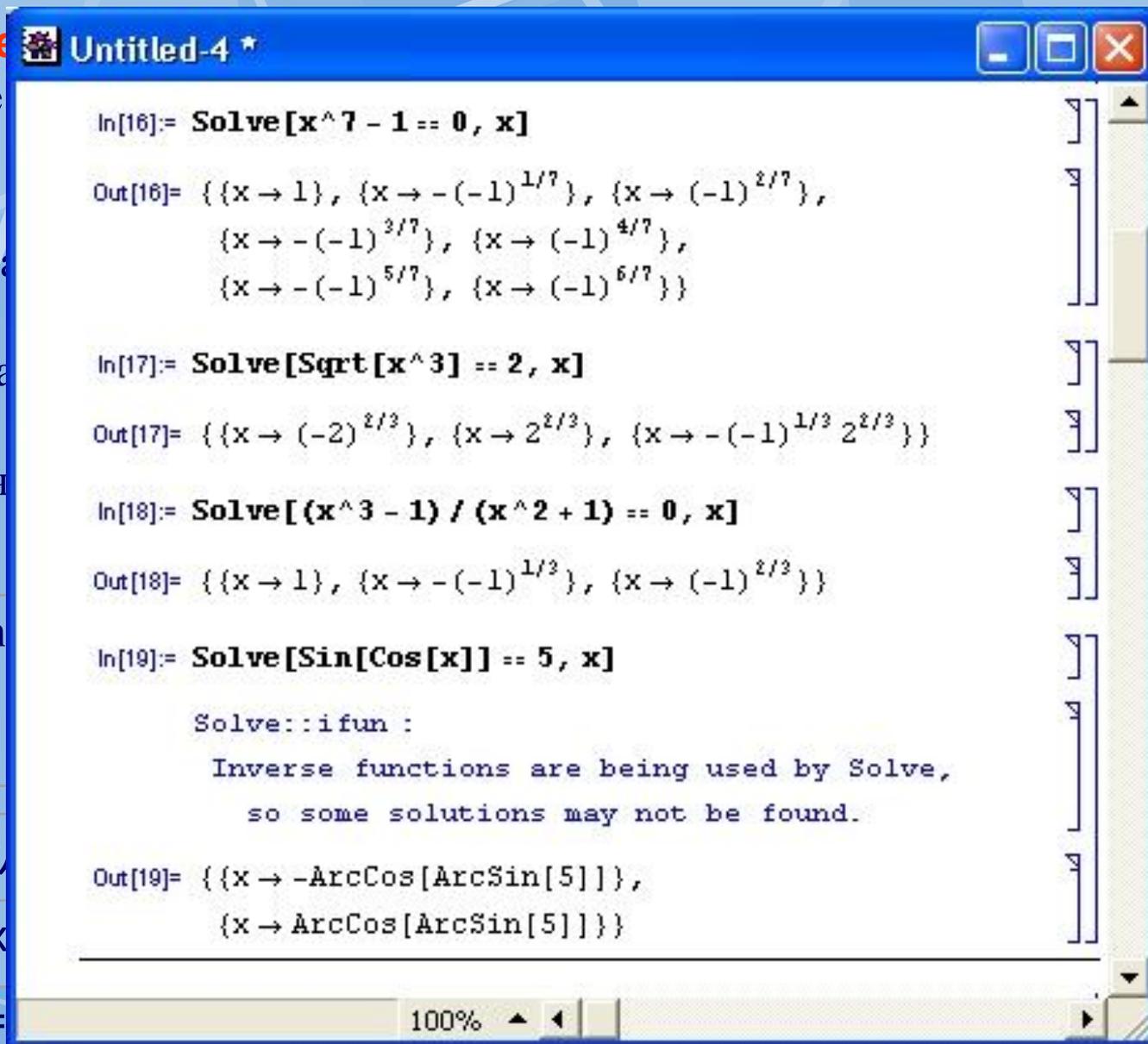
vars.

Solve[eqn

Ввод (In,

Solve[a*x

Solve[{y=



```
In[16]:= Solve[x^7 - 1 == 0, x]
Out[16]= {{x -> 1}, {x -> -(-1)^(1/7)}, {x -> (-1)^(2/7)},
          {x -> -(-1)^(3/7)}, {x -> (-1)^(4/7)},
          {x -> -(-1)^(5/7)}, {x -> (-1)^(6/7)}}

In[17]:= Solve[Sqrt[x^3] == 2, x]
Out[17]= {{x -> (-2)^(2/3)}, {x -> 2^(2/3)}, {x -> -(-1)^(1/3) 2^(2/3)}}

In[18]:= Solve[(x^3 - 1) / (x^2 + 1) == 0, x]
Out[18]= {{x -> 1}, {x -> -(-1)^(1/3)}, {x -> (-1)^(2/3)}}

In[19]:= Solve[Sin[Cos[x]] == 5, x]
Solve::ifun :
  Inverse functions are being used by Solve,
  so some solutions may not be found.
Out[19]= {{x -> -ArcCos[ArcSin[5]]},
          {x -> ArcCos[ArcSin[5]]}}
```

уравнений

уравнения).

внений. Для

переменным

уравнений

пючением

Расширенные математические возможности

Число

принципе
решение
используется

`NSolve[eq`

`NSolve[eq`

Ввод (

`NSolve`

`NSolve`

```
In[68]:= f[x_] = 2 * x^2 - 5 * x + 1
Out[68]= 1 - 5 x + 2 x^2
In[69]:= Plot[f[x], {x, -5, 6}]
Out[69]= - Graphics -
In[70]:= NSolve[f[x] == 0, x]
Out[70]= {{x -> 0.}, {x -> 3. - 3.4641 i}, {x -> 3. + 3.4641 i}}
```

Solve

равнений в
возможно их
уравнений

ОМ ОДНО
СИТЕЛЬНО

МЕТОДОМ
IX vars,

Расширенные математические возможности

Решение дифференциальных уравнений в символьном виде

Дифференциальными уравнениями принято называть уравнения, в состав которых входят производные функции $y(x)$, представляющей решение уравнения. Дифференциальные уравнения могут быть представлены в различной форме, например в общеизвестной форме Коши

$$y'(x) = \text{eqn} = f(x, y)$$

Несколько взаимосвязанных дифференциальных уравнений образуют систему.

Для решения дифференциальных уравнений в символьном виде используются следующие средства:

<code>DSolve[eqn,y[x],x]</code>	- решает дифференциальное уравнение относительно функции $y[x]$ с независимой переменной x ;
<code>DSolve[{eqn1,eqn2,...}, {y1[x1,...],...},{x1,...}]</code>	- решает систему дифференциальных уравнений;

Расширенные математические возможности

Приведем примеры решения дифференциальных уравнений:

Ввод (<i>In</i>)	Вывод (<i>Out</i>)
<code>DSolve[y'[x]==2*a*x^3,y[x],x]</code>	$\left\{ \left\{ y[x] \rightarrow -1 + x \frac{C[1]}{E^x} \right\} \right\}$
<code>DSolve[y'[x]+y[x]==x,y[x],x]</code>	$\left\{ \left\{ y[x] \rightarrow \frac{ax^4}{2} + C[1] \right\} \right\}$

Расширенные математические возможности

Решение дифференциальных уравнений в численном виде

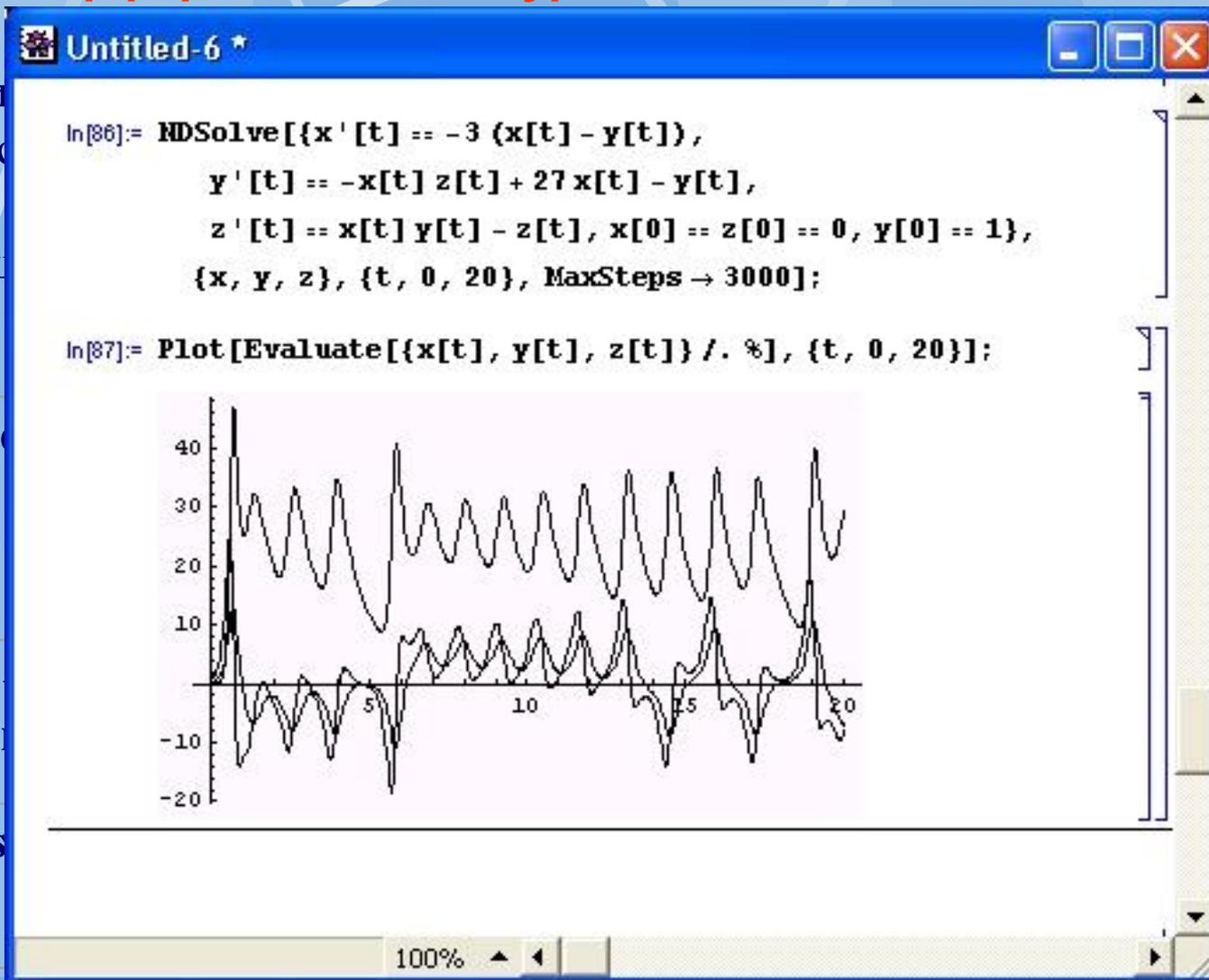
Многие
аналитические
решены
дифференци-

не имеют
могут быть
систем

`NDSolve[`

`NDSolve[`
`{x,xmin,x`

`MaxSteps`



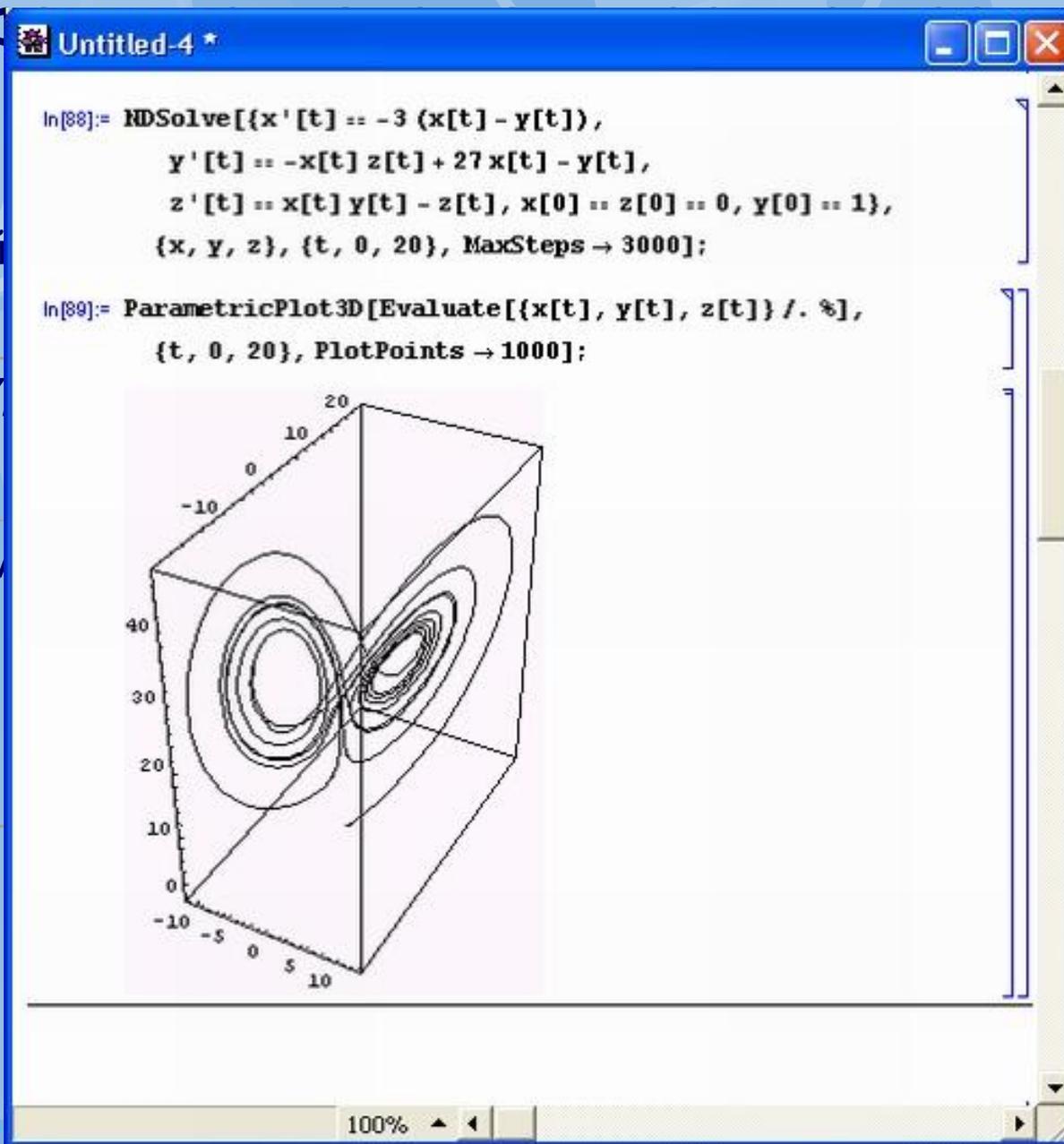
дифферен-
циаль-
ных урав-
нениях.
числен-
ных

Приведем
уравнений

ЦИАЛЬНЫХ

Ввод (

NDSolve



Расширенные математические возможности

Функции минимизации и максимизации

В практике прикладных математических вычислений важная роль принадлежит оптимизационным задачам, например таким, как поиск минимальных и максимальных значений функции одной или ряда переменных. Система Mathematica дает разнообразные возможности по решению задач оптимизации – от поиска элементов списка с минимальным или максимальным значением до поиска локальных или даже глобальных минимумов функции, заданных аналитически.

Для поиска максимального и минимального значений ряда чисел, входящих в список имеются следующие функции:

$\text{Max}[x_1, x_2, \dots]$	- возвращает наибольшее из x_i ;
$\text{Max}[\{x_1, x_2, \dots\}, \{y_1, \dots\}, \dots]$	- возвращает наибольший элемент любого из списков;
$\text{Min}[x_1, x_2, \dots]$	- возвращает наименьшее из x_i ;
$\text{Min}[\{x_1, x_2, \dots\}, \{y_1, \dots\}, \dots]$	- возвращает наименьший элемент любого из списков;

Расширенные математические возможности

Приведенные ниже примеры показывают действие этих простых функций:

Ввод (<i>In</i>)	Вывод (<i>Out</i>)
$\text{Max}[1,5,2,6.5,3,4]$	6.5
$\text{Max}[\{1,3,2\},\{4,5,6\},\{9,8,7\}]$	9
$\text{Min}[1,5,2,6.5,-3,4]$	-3
$\text{Min}[\{1,3,2\},\{4,5,6\},\{9,8,7\}]$	1

Расширенные математические возможности

Для поиска локального минимума некоторой аналитической функции $f(x)$ используется функция:

FindMinimum[f,{x,x0}]

- выполняет поиск локального минимума f начиная со значения $x=x_0$ и возвращает его значение со значением x в точке минимума.

Приведем примеры применения функции FindMinimum :

Ввод (In)

Вывод (Out)

FindMinimum[-x*Exp[-2*x],
{x,1}]

{-0.18394,{x->0.5}}

FindMinimum{-5*x*Exp[-x/2]*(2+Sin
[3*x]),{x,4}}

{-6.79134,{x->4.6179}}

Расширенные математические возможности

Для поиска глобального максимума и минимума аналитически заданной функции ряда переменных служат следующие две функции:

ConstrainedMax[f, {inequalities}, {x,y,...}]

- ищет глобальный максимум функции f в области, определяемой неравенствами $inequalities$. Предполагается, что все переменные x, y, \dots не отрицательны.

ConstrainedMin[f, {inequalities}, {x,y,...}]

- ищет глобальный минимум функции f в области, определяемой неравенствами $inequalities$. Все переменные x, y, \dots полагаются не отрицательными.

Эти функции решают типовые задачи линейного программирования. В дополнение к ним может быть использована функция:

LinearProgramming[c,m,b]

- ищет вектор x , минимизирующий величину $c \cdot x$ в соответствии с условиями $m \cdot x \geq b$ и $x \geq 0$.

Расширенные математические возможности

К задачам на минимизацию относятся также задачи линейной и нелинейной регрессии. В них вычисляются параметры некоторой функции, при которых среднеквадратичная погрешность между результатами вычислений по этой функции и совокупностью исходных данных минимальна. В отличие от интерполяции при регрессии данная функция в узловых точках не дает точного значения ординат – она просто минимизирует в них погрешности вычислений.

Для решения таких задач используется функция **Fit**.

Fit[data,func,vars]

Для списка данных `data` она ищет приближения методом наименьших квадратов в виде линейной комбинации функций `funcs` переменных `vars`. Данные `data` могут иметь форму $\{\{x_1, y_1, \dots, f_1\}, \{x_2, y_2, \dots, f_2\}, \dots\}$, где число координат x, y, \dots равно числу переменных в списке `vars`. Они могут быть представлены также в форме $\{f_1, f_2, \dots\}$ с одной координатой, принимающей значения $1, 2, \dots$. Аргумент `funcs` может быть любым списком функций, которые зависят только от объектов `vars`.

Расширенные математические возможности

Следующие примеры показывают приближение исходных данных степенным полиномом и линейной комбинацией двух функций:

Ввод (In)	Вывод (Out)
Fit[{{0,0.9},{2,8.1}, {3,17},{4,33}}, {a,x,x^2},x]	0.997273-1.40864x+2.33409x ²
Fit[{{0,0.9},{2,8.1}, {3,17}}, {x^2,Exp[x],x},x]	0.9E ^x + 2.89276 x - 1.08392 x ²

Расширенные математические возможности

Символьные операции. Работа с выражениями.

Одним из важнейших понятий системы является математическое выражение, или просто выражение - expr (от слова expression). Выражение может быть представлено в общепринятом виде (как математическая формула или ее часть) с помощью операторов, например: $a*(x+y+z)$ или x^y ; оно может задавать и некоторую функцию $f[x,y,...]$. Наряду с такой формой существует так называемая полная форма представления выражений, при которой основные арифметические операции задаются не операторами, а только соответствующими функциями. Фактически именно с выражениями, представленными в полной форме, оперирует символьное ядро системы.

В системе предусматривается различная работа с выражениями (сложение, умножение, возведение в степень, создание списка и т.д.

Расширенные математические возможности

Упрощение выражений

Упрощение математических выражений - одна из самых важных задач символической математики. Порою невероятно сложное математическое выражение (которое пугает с первого вида) является просто нулем или единицей либо сводится к простому выражению после ряда преобразований. Качество выполнения операции упрощения во многом определяется мощностью ядра математической системы, поскольку зависит от числа функций и правил преобразования выражений, заложенных в ее символическое ядро.

Для упрощения выражений используется функция `Simplify`:

<code>Simplify[expr]</code>	- исполняет последовательность алгебраических преобразований над выражением <code>expr</code> и возвращает простейшую из найденных форм.
------------------------------------	--

Ввод (In)	Вывод (Out)
<code>Simplify[6*x^2]</code>	12x
<code>Simplify[x^3*y*x^5]</code>	x^8y

Расширенные математические возможности

Функции раскрытия и расширения выражений

Расширение или раскрытие выражений - еще одна типовая операция компьютерной алгебры. Часто компактная форма представления выражений обусловлена определенными операциями по их упрощению. Существует множество выражений, для которых эти правила известны.

Например:

Ввод (<i>In</i>)	Вывод (<i>Out</i>)
$(a-b)^2$	$a^2-2ab+b^2$

Расширенные математические возможности

Основные функции, возвращающие результаты с раскрытием и расширением выражений:

ComplexExpand[expr]	- раскрывает expr, полагая все переменные вещественными;
ComplexExpand [expr, {x1,x2,...}]	- раскрывает expr, полагая переменные соответствующими какому-либо действительному x_i или комплексному;
Expand[expr]	- раскрывает произведения и положительные целые степени в expr;
ExpandAll[expr]	- раскрывает все произведения и целочисленные степени в любой части expr;
ExpandDenominator[expr]	- раскрывает произведение и степени, которые присутствуют в выражении expr в роли знаменателей;
ExpandNumerator[expr]	- раскрывает произведение и степени, которые присутствуют в выражении expr в роли числителей;
PowerExpand[expr]	- раскрывает вложенные степени, степени произведений, логарифмы от степеней и логарифмы от произведений.

Расширенные математические возможности

Примеры:

Ввод (<i>In</i>)	Вывод (<i>Out</i>)
<code>Expand[(x-a)*(x-b)*(x-c)]</code>	$-(abc)+abx+acx+bcx-ax^2-bx^2-cx^2+x^3$
<code>ExpandNumerator[(1+x)^2/x]</code>	$\frac{1 + 2x + x^2}{x}$

Расширенные математические возможности

К операциям, расширяющим выражения относятся также функции:

Collect [expr,x]	- выполняет приведение общих членов выражения по степеням переменной x;
Collect [expr,{x1,x2,...}]	- выполняет приведение общих членов выражения по степеням переменных x1,x2....

Примеры:

Ввод (In)	Вывод (Out)
$\text{Collect}[(x-1)(x-2)(x^2-9),x]$	$-18+27x-7x^2-3x^3+x^4$
$\text{Collect}[(x-y)^3x+(y-x)y,\{x,y\}]$	$x^4-3x^3y+y^2+3x^2y^2+x(-y-y^3)$

Расширенные математические возможности

Функции для расширенных операций с выражениями

Мы рассмотрели немногочисленную группу функций для работы с выражениями: их упрощения, расширения, выделения множителей и т.д. Эти функции способны решать большинство повседневных задач, связанных с аналитическими преобразованиями выражений. Однако система имеет гораздо более полный набор функций для работы с выражениями.

Для расширенной работы с выражениями служат следующие функции:

Apart[expr]	- возвращает expr, записывая заново рациональное выражение как сумму членов с минимальным знаменателями;
Cancel[expr]	- возвращает expr, выполняя сокращение общих множителей числителя и знаменателя;
Denominator[expr]	- возвращает знаменатель выражения;
Numerator[expr]	- возвращает числитель выражения;
Exponent[expr,form]	- возвращает максимальную степень, с которой form присутствует в expr;
Together[expr]	- приводит члены суммы к общему знаменателю и сокращает множители в полученном результате.

Расширенные математические возможности

Примеры:

Ввод (<i>In</i>)	Вывод (<i>Out</i>)
<code>Apart[(x^4+1)/(x^2-1)]</code>	$1 + \frac{1}{-1+x} + x^2 - \frac{1}{1+x}$
<code>Cancel[(x^2-1)/(x-1)]</code>	$1+x$
<code>Numerator[(x^2-x-1)/(x-1)]</code>	$-1-x+x^2$
<code>Denominator[(x^2-x-1)/(x-1)]</code>	$-1+x$

Графические возможности пакета Mathematica

Графика всегда была козырной картой системы. Графические возможности достигаются как обилием встроенных функций, так и средствами их модификации с помощью директив, опций и примитивов. Система позволяет строить практически любые виды математических графиков, причем обычного пользователя в большинстве случаев удовлетворяют графики, параметры которых система задает по умолчанию.

Начнем рассмотрение графических возможностей системы с построения графиков функций одной переменной вида $y = f(x)$ или просто $f(x)$. График таких функций строится на плоскости, т.е. в двумерном пространстве. Он представляет собой геометрическое место точек (y, x) при изменении независимой переменной x (абциссы) в заданных пределах, например от минимального значения x_{\min} до максимального значения x_{\max} .

Графические возможности пакета Mathematica

Для построения двумерных графиков функций вида $f(x)$ используется встроенная в ядро функция Plot. Она задается в следующих формах:

Plot[f,{x,xmin,xmax}]	- строит график функции f аргумента x в интервале от x_{\min} до x_{\max} .
Plot[{f1,f2,...},{x,xmin,xmax}]	- строит графики ряда функций f_i ($i = 1, 2, \dots$).

Формат построенных графиков задается специальными графическими опциями и директивами.

С функцией Plot используются различные опции, меняющие те или иные параметры графиков и их вид (например, размер графика, наличие осей и т.д.) Опции внутри записей графических функций задаются своим именем name и значением value в виде: name-> value

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

Automatic - используется автоматический выбор;

None - опция не используется;

All - используется в любом случае.

Графические возможности пакета Mathematica

Поскольку графики являются объектами, то они могут быть значениями переменных. Поэтому Mathematica допускает следующие конструкции:

`Plot[Sin[x],{x,0,20}]` - построение графика синусоиды;

`g:=Plot [Sin [x], {x, 0, 20}]` - задание объекта - графика синусоиды
- с отложенным выводом;

`g=Plot [Sin [x], {x, 0, 20}]` - задание объекта - графика синусоиды
- с немедленным выводом.

Графические возможности пакета Mathematica

Опции функции Plot

С функцией Plot используются различные опции, меняющие те или иные параметры графиков и их вид (например, размер графика, наличие осей и т.д.). Опции внутри записей графических функций задаются своим именем name и значением value в виде:

name -> value

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

Autonomic – используется автоматический выбор;

None – опция не используется;

All – используется в любом случае;

True – используется;

False – не используется.

Графические возможности пакета Mathematica

Ниже представлены основные опции для графических функций (звездочкой отмечены те опции, которые можно использовать и в трехмерной графике):

\$ AspectRatio*	- задает пропорцию размеров графика - отношение высоты к ширине;
Axes*	- устанавливает, должны ли рисоваться оси (по умолчанию Automatic);
AxesLabel*	- устанавливает, нужны ли отметки на осях графиков (по умолчанию None);
AxesOrigin	- указывает, где должно располагаться начало отсчета осей (по умолчанию Automatic);
AxesStyle*	- указывает в каком стиле должны строиться оси графика (по умолчанию Automatic);
Background*	- указывает какого цвета должен быть фон графика (по умолчанию – Automatic);
\$DisplayFunction*	- указывает канал и порядок вывода графического объекта (по умолчанию -\$DisplayFunction);
PlotRange	- указывает какие точки включать в график (по умолчанию Automatic);
PlotLabel*	- указывает нужно ли вывести титульную надпись для графика (по умолчанию None);
\$ PlotStyle	- устанавливает стиль отображения графика (по умолчанию Automatic);
RotateLabel	- указывает, следует ли метки на вертикальных осях развернуть так, чтобы они стали вертикальными;
Frame	- указывает, нужно ли рисовать рамку вокруг графика (по умолчанию False).
Ticks*	- устанавливает штриховые метки для осей (по умолчанию Automatic);

Графические возможности пакета Mathematica

Опции могут серьезно изменить вид графика.

Примеры использования опций графики:

Установка масштаба 2D-графика по оси y

```
Plot[Sin[x]/x,{x,-20,20},PlotRange->{-0.25,1}]
```

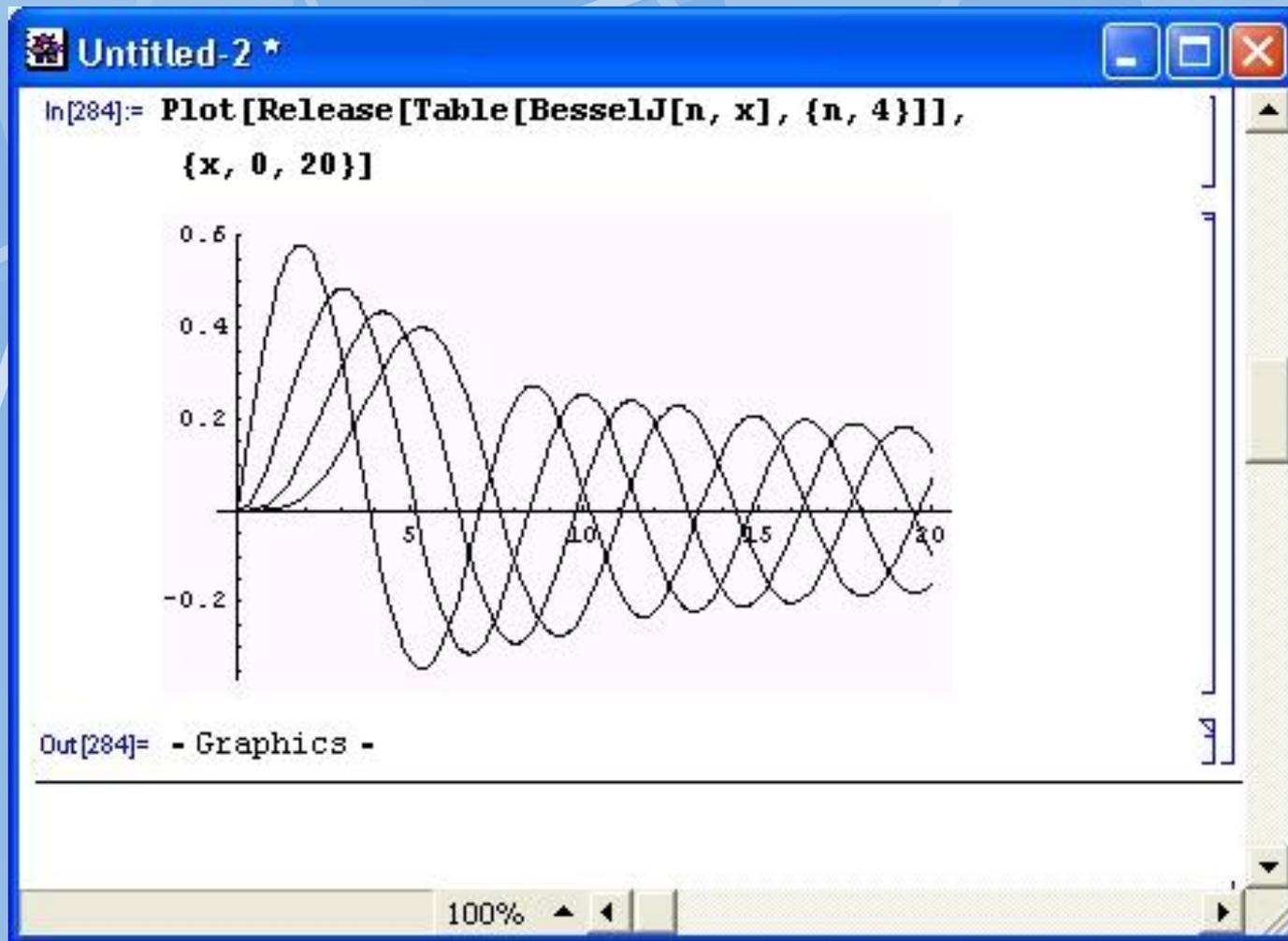
Ввод надписей по осям 2D – графика

```
Plot[Sin[x]^3,{x,0,20},AxesLabel->{"x value", "Graphic Sin(x)^3"}]
```

График с титульной надписью

```
Plot[Sin[x]^3,{x,0,20},Axes->None,PlotLabel->"Graphic for functions Sin(x)^3"]
```

Графические возможности пакета Mathematica



Если желательно выделение линий разными цветами, удобно использовать в качестве значения опции `PlotStyle` список вида `{Hue [c1] , Hue [c2] , ...}`, где параметры c_1, c_2, \dots выбираются от 0 до 1 и задают цвет соответствующей кривой.

Графические возможности пакета Mathematica

Директивы двумерной графики

Еще одним мощным средством графики системы являются графические директивы. Они указывают, с какими графическими параметрами (цвет, толщина линий, их стиль и т.д.) должна строить графики та или иная функция.

В системе используются следующие директивы двумерной графики:

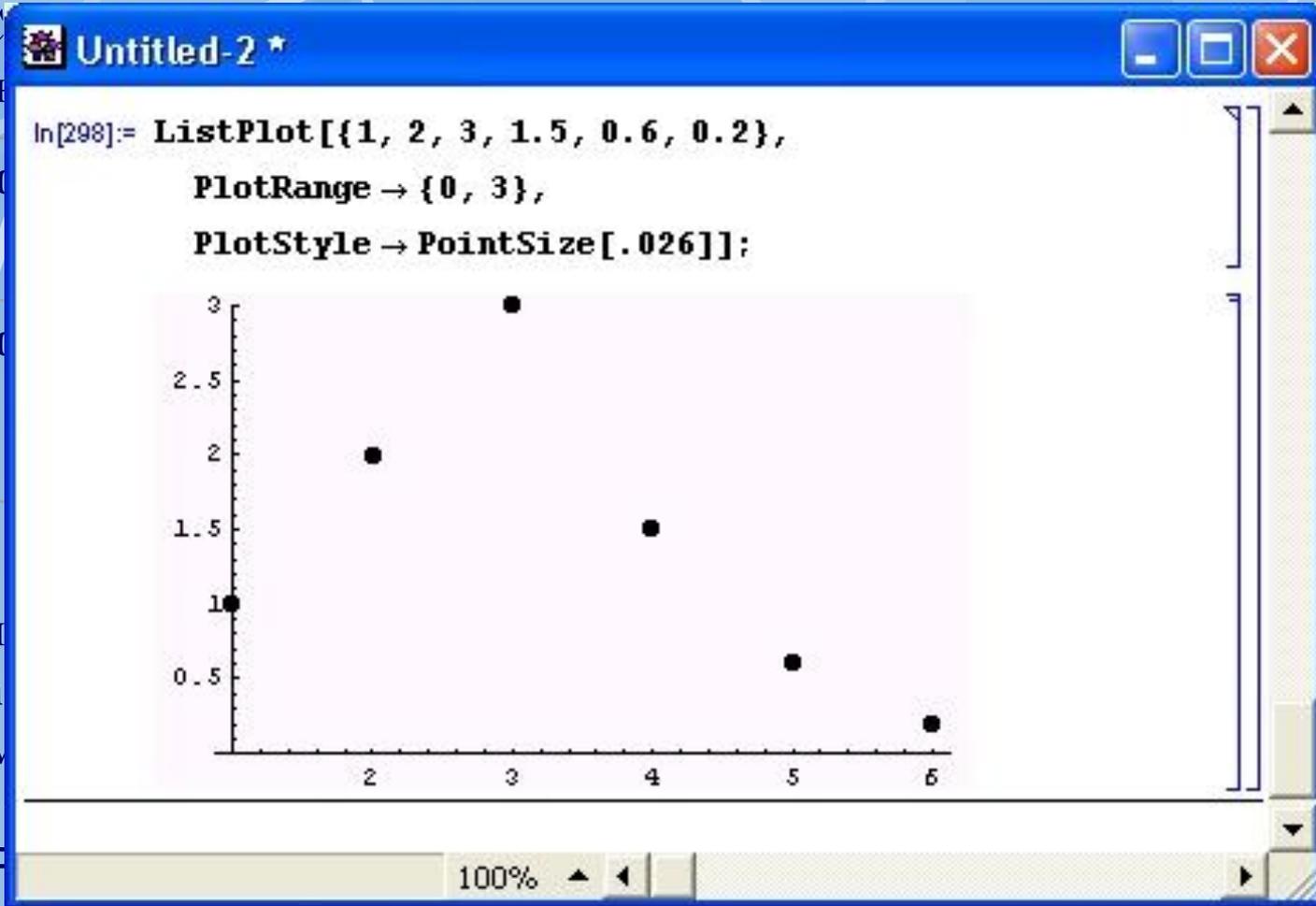
AbsoluteDashing[{d1,d2,...}]	- указывает, что последующие линии должны рисоваться пунктиром со смежными (последовательными) сегментами, имеющими абсолютные длины d1,d2 и повторяющимися циклически (значения длины di задаются в пикселях);
AbsolutePointSize[d]	- устанавливает построение точек графика в виде кругов с абсолютным радиусом d (в пикселях);
AbsoluteThickness[d]	- устанавливает толщину d рисуемых линий;
Dashing[{r1,r2,...}]	- устанавливает вывод линий пунктиром с последовательными сегментами длиной r1, r2,... повторяющимися циклически;
Thickness[r]	- устанавливает толщину r для всех последующих линий r.

Пример использования директив двумерной графики:

```
Plot[Sin[x],{x,0,20},PlotStyle-> Dashing[{0.05,0.025}]]
```

Графические возможности пакета Mathematica

Графическая функция ListPlot



С
встроенн

ListPl

ListPl

Отм
PlotJoin
отрезкам
Пример
g = ListP

существует

1, y2, ... и

соединять

Графические возможности пакета Mathematica

Получение информации о графических объектах

Информацию об опциях графического объекта g дают следующие функции:

`FullAxes [g]` — возвращает список опций координатных осей;

`Options [g]` - возвращает упрощенный список опций;

`FullOptions [g]` - возвращает полный список опций;

`InputForm[g]` - возвращает информацию о графике (включая таблицу точек).

Пусть задан графический объект g : `g:=Plot[Sin[x],{x,-10,10}]`

Ниже представлено получение упрощенного списка опций этого графического объекта:

```
Options[g]
```

```
{PlotRange -> Automatic, AspectRatio -> 1/GoldenRatio,
```

```
DisplayFunction -> $DisplayFunction, ColorOutput -> Automatic, Axes -> Automatic,
```

```
AxesOrigin -> Automatic, PlotLabel -> None, AxesLabel -> None, Ticks -> Automatic,
```

```
GridLines -> None, Prolog -> {}, Epilog -> {}, AxesStyle -> Automatic, Background ->
```

```
Automatic, DefaultColor -> Automatic, DefaultFont -> $DefaultFont, RotateLabel ->
```

```
True, Frame -> False, FrameStyle -> Automatic, FrameTicks -> Automatic!,
```

```
FrameLabel -> None, PlotRegion -> Automatic, ImageSize -> Automatic, TextStyle ->
```

```
$TextStyle, FormatType -> $FormatType}
```

Графические возможности пакета Mathematica

Перестройка и комбинирование графиков

При построении графиков приходится изменять их вид, а также те или иные параметры и опции. Для этого можно повторить вычисления, но тогда заметно снизится скорость работы с системой. Для ее повышения удобно использовать специальные функции перестройки и вывода графиков, учитывающих что узловые точки этих графиков уже были рассчитаны и их координаты хранятся в памяти ПК.

В этом случае удобно использовать следующие функции:

<code>\$ Show[plot]</code>	- построение графика по вычисленным данным;
<code>\$ Show[plot,option->value]</code>	- построение графика с заданной опцией;
<code>\$ Show[plot1,plot2,...]</code>	- построение нескольких графиков с наложением их друг на друга.

Примеры: *Построение объединенного сдвоенного 2D-графика*

```
g1 = Plot[Sin[x],{x,0,20}];
```

```
g2 = Plot[Sin[x]^3,{x,0,20}];
```

```
Show[g1,g2,PlotRange->{-1.5,1.5}]
```

Графические возможности пакета Mathematica

Примитивы двумерной графики

Примитивами двумерной графики называют дополнительные указания, вводимые в функцию `Graphics`, для построения с ее помощью некоторых геометрических фигур. Сама функция `Graphics` задается в следующем виде:

`Graphics[primitives,options]`

Применение примитивов в составе функции `Graphics` избавляет пользователя от задания довольно сложных математических выражений, описывающих эти фигуры, и описания алгоритмов построения по ним графиков фигур. Примитивы могут дополнять и иные функции.

Графические возможности пакета Mathematica

Circle[{x,y},r]	- окружность радиуса r с центром $\{x,y\}$;
Circle[{x,y},{rx,ry}]	- эллипс с полуосями r_x и r_y с центром $\{x,y\}$;
Circle[{x,y},r,{theta1,theta2}]	- дугу окружности радиуса r с углами конечных точек θ_1 и θ_2 с центром $\{x,y\}$;
Disk[{x,y},r]	- закрашенный круг радиусом r с центром $\{x,y\}$;
Disk[{x,y},{rx,ry}]	- закрашенный полуовал с полуосями r_x и r_y с центром $\{x,y\}$;
Disk[{x,y},r,{theta1,theta2}]	- сегмент круга с углами конечных точек образующей окружности θ_1 и θ_2 , радиусом r и с координатами центра $\{x,y\}$;
Line[{{x1,y1},{x2,y2},...}]*	линию в виде отрезков прямых, соединяющих точки с координатами $\{x_1,y_1\}$, $\{x_2,y_2\}$,
Point[{x,y}]*	- точку с координатами $\{x,y\}$.
Text[expr,coords]	- выражение $expr$ в текстовой форме с центром, имеющим координаты $coords$.

Графические возможности пакета Mathematica

Пример

Построение
текста

$g1 = G$

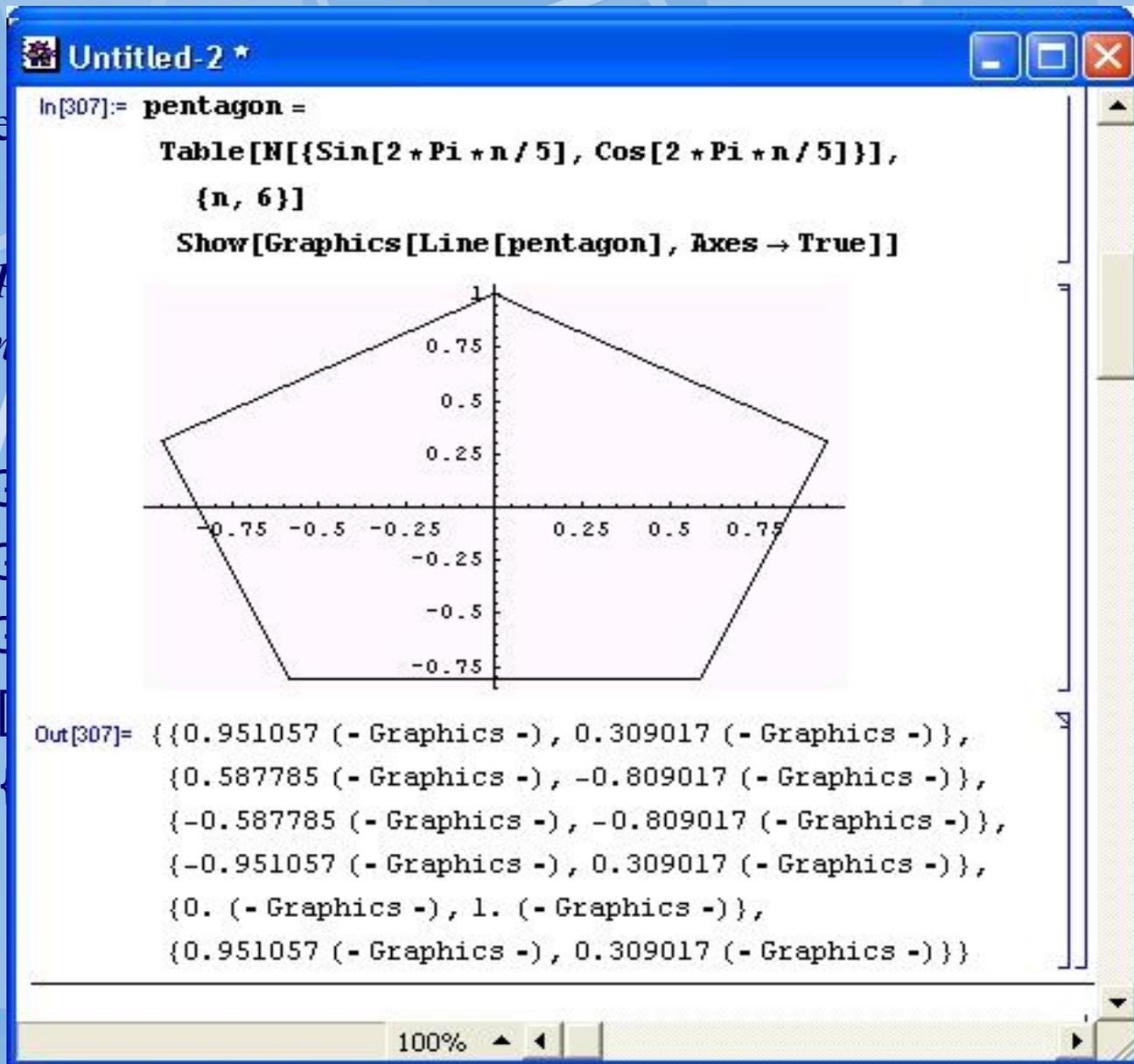
$g2 = G$

$g3 = G$

Show[

Point[

кругности,



Графические возможности пакета Mathematica

Построение графиков параметрически заданных функций

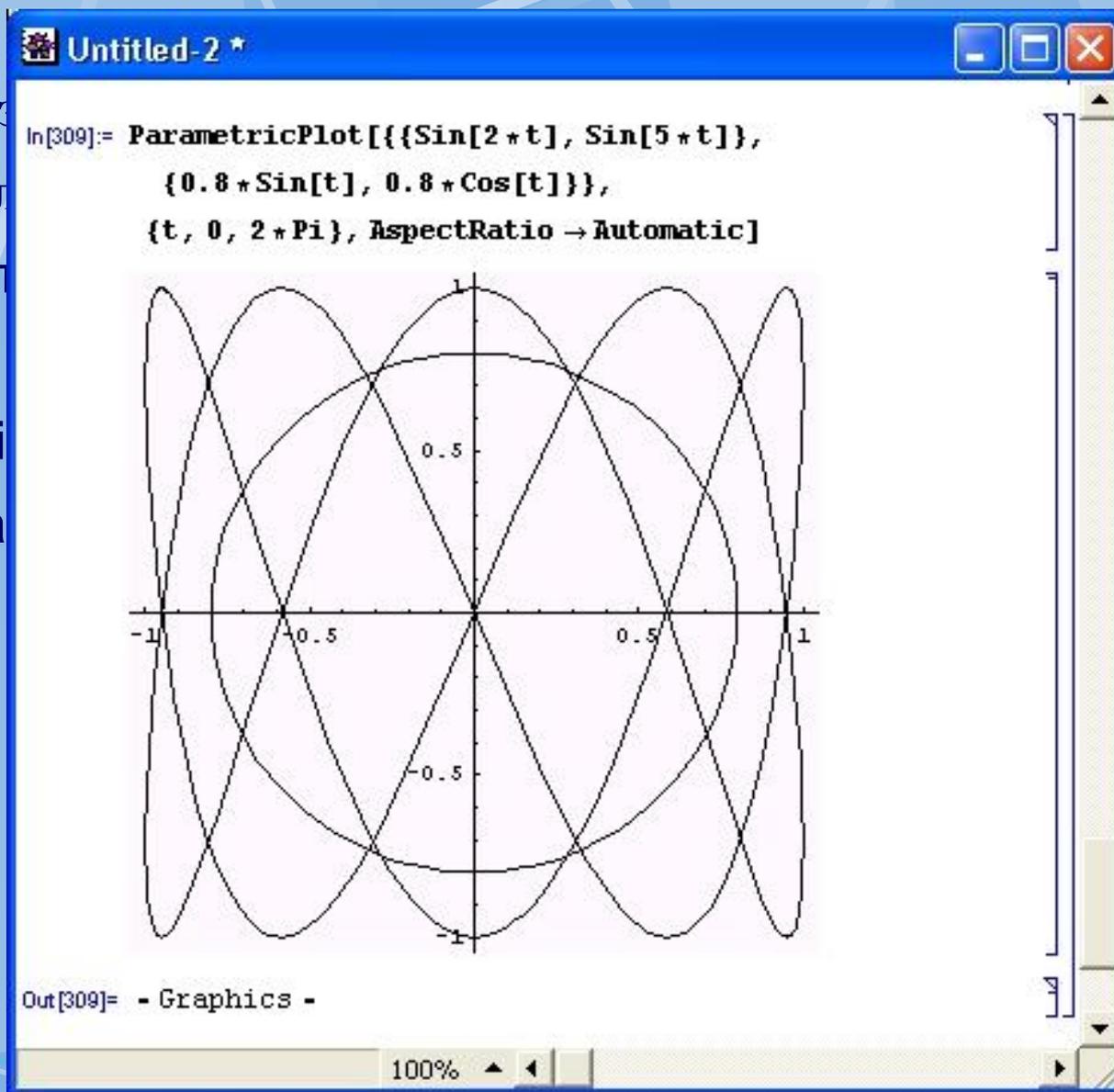
Функции, графики которых строятся на плоскости, могут быть заданы в параметрическом виде: $x=f_x(t)$ и $y=f_y(t)$, где независимая переменная t меняется от минимального значения t_{\min} до максимального t_{\max} с шагом dt . Особенно удобно применение таких функций для построения замкнутых линий, таких как окружности, эллипсы, циклоиды и др.

Для построения параметрически заданных функций используются следующие графические функции системы:

ParametricPlot {{fx,fy},{t,tmin,tmax}}	- строит параметрический график с координатами f_x и f_y , получаемыми как функции от t ;
ParametricPlot {{fx,fy}, {gx,gy},{t,tmin,tmax}}	- строит графики нескольких параметрических кривых.

Графические возможности пакета Mathematica

Для из
что позвол
Пример п
Parametri
AspectRa



ВЫ И ОПЦИИ,
В.
СКИ:

Графические возможности пакета Mathematica

Построение графиков трехмерных поверхностей.

Функция двух переменных $z = f(x,y)$ в пространстве образует некоторую трехмерную поверхность или фигуру. Для их построения приходится использовать координатную систему с тремя осями координат: x , y , z .

Для построения графиков трехмерных поверхностей используются следующие основные графические функции:

ListPlot3D[array]

- строит трехмерный график поверхности, представленной массивом значений высот array;

**Plot3D[f[x,y],{x,xmin,xmax},
{y,ymin,ymax}]**

- строит трехмерный график для аналитически заданной функции f переменных x, y , изменяющихся в заданных пределах;

Графические возможности пакета Mathematica

На рис. показан пример построения поверхности, описываемой функцией двух переменных $\cos(x+y)$ при x и y , меняющихся от -3 до 3 . Поверхность строится в виде каркаса с прямоугольными ячейками с использованием функциональной окраски. Все опции заданы по умолчанию.

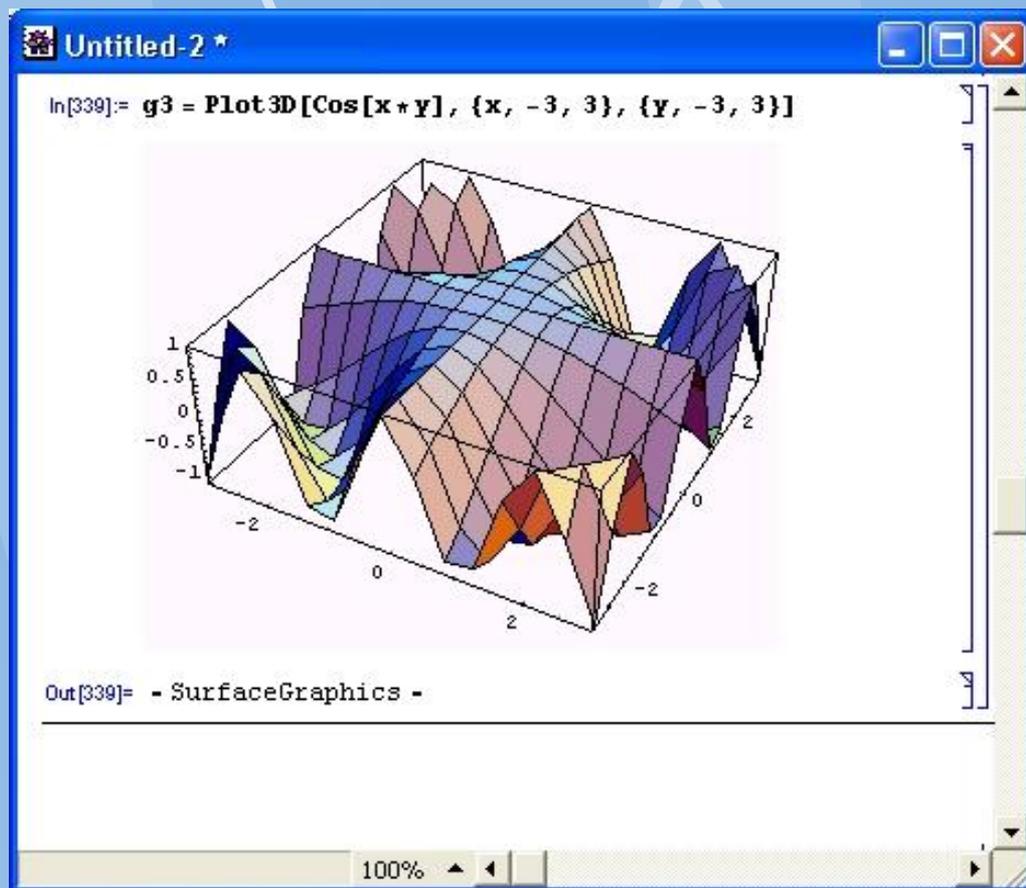


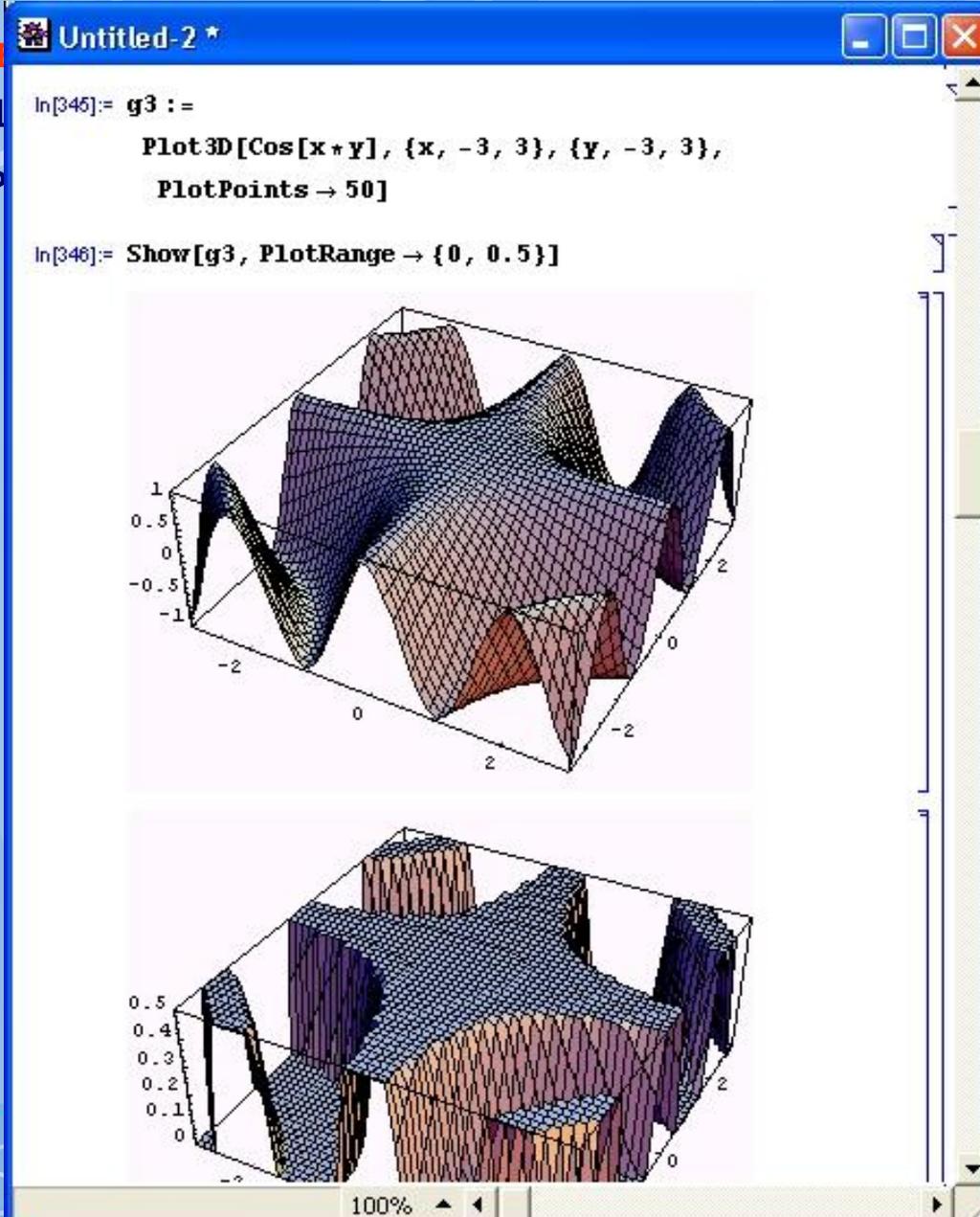
Рис. Пример построения поверхности $\cos(xy)$ функцией `Plot3D` с опциями по умолчанию

Графические возможности пакета Mathematica

Опции и директивы

Для модификации
многочисленны

пользоваться



The screenshot shows a Mathematica window titled "Untitled-2 *". The input area contains the following code:

```
In[345]:= g3 :=  
    Plot3D[Cos[x+y], {x, -3, 3}, {y, -3, 3},  
    PlotPoints -> 50]  
  
In[348]:= Show[g3, PlotRange -> {0, 0.5}]
```

The output consists of two 3D plots. The top plot shows the surface $z = \cos(x+y)$ over the domain $x \in [-3, 3]$ and $y \in [-3, 3]$. The surface is rendered with a grid mesh and a color gradient from blue to red. The axes are labeled with values -2, 0, and 2. The bottom plot is a zoomed-in view of the same surface, showing the grid lines more clearly. The vertical axis (z) is labeled with values 0, 0.1, 0.2, 0.3, 0.4, and 0.5. The horizontal axes are also labeled with values -2, 0, and 2. The bottom status bar shows "100%" zoom level.

Графические возможности пакета Mathematica

Для построения трехмерных графиков **в параметрической форме** также имеются функции:

ParametricPlot3D[fx,fy,fz]	- строит трехмерную поверхность, параметризованную по t и u в заданных пределах их изменения.
ParametricPlot3D[fx,fy,fz}, {t,tmin,tmax}]	- выполняет построение трехмерной кривой, параметризованной по переменной t , которая изменяется от $tmin$ до $tmax$.
и другие.	

Примеры построения трехмерных графиков:

Исходная математическая поверхность

```
g3=Plot3D[Cos[x*y],{x,-3,3},{y,-3,3},PlotPoints->40]
```

Математическая поверхность с отсеченной верхней частью

```
Show[g3,PlotRange->{0,0.5}]
```

Графические возможности пакета Mathematica

Примитивы трехмерной графики

Наряду с построением графиков поверхностей, заданных аналитическими выражениями, имеется возможность создания графиков различных объектов с помощью функции

Graphics3D[primitives,options] - представляет трехмерное графическое изображение.

Основные примитивы:

Cuboid[{xmin,ymin,zmin}]

- строит единичный куб с гранями, ориентированными параллельно осям};

**Cuboid[{xmin,ymin,zmin},
{xmax,ymax,zmax},]**

- строит прямоугольный параллелепипед, заданный координатами противоположных вершин.

Графические возм

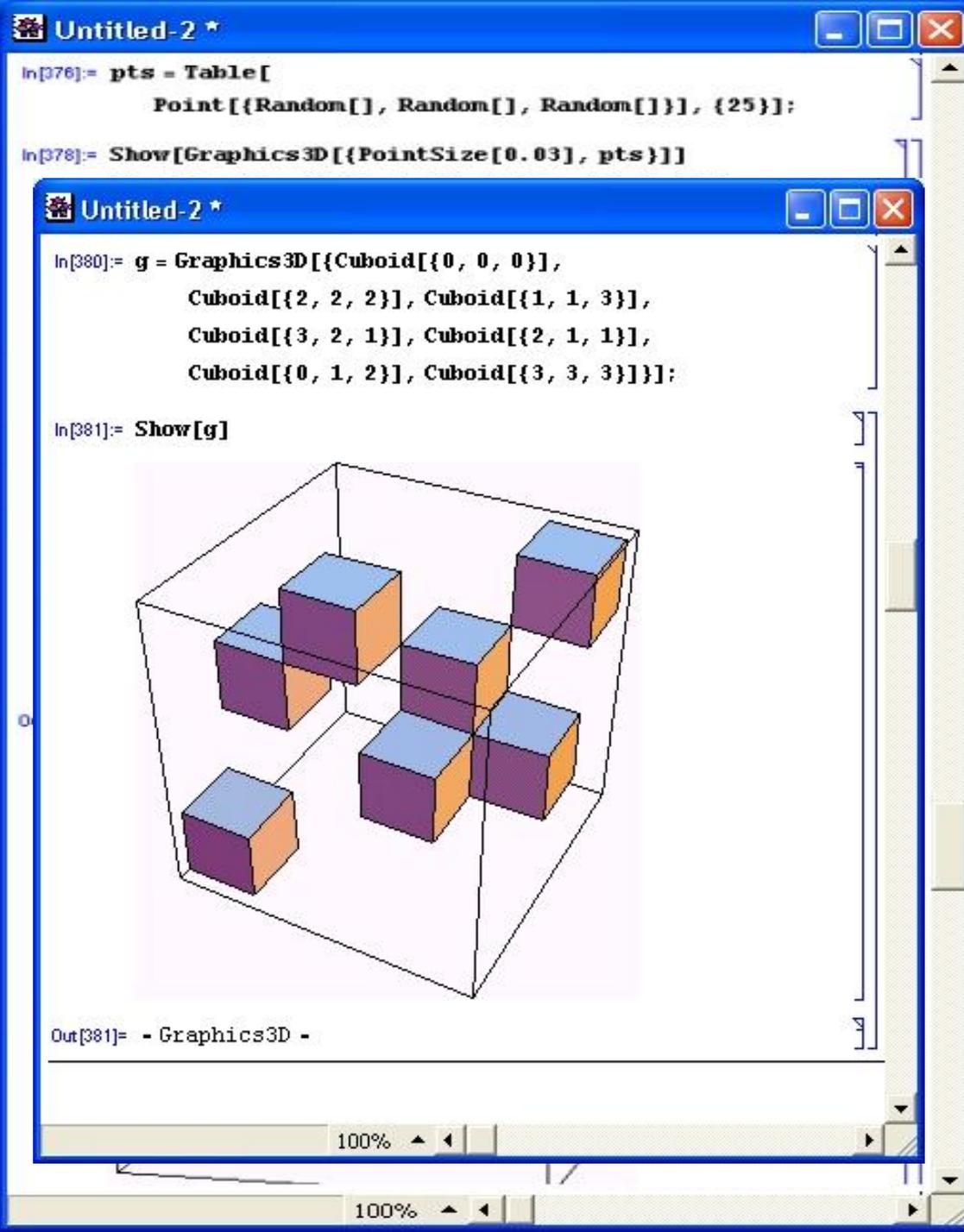
Примеры:

Построение куба с точками

```
pts=Table[Point[{Random[],Random[],Random[]}, {25}];  
Show[Graphics3D[{PointSize[0.03], pts}]]  
Show[Graphics3D[pts],ViewPoint->{1,1,1}]
```

Построение шести кубов в

```
g=Graphics3D[{Cuboid[{0,0,0}, {1,1,1}],  
Cuboid[{1,1,3}],Cuboid[{2,2,2}],  
Cuboid[{3,2,1}],Cuboid[{2,1,1}],  
Cuboid[{0,1,2}],Cuboid[{3,3,3}]}];  
Show[g]
```



Графические возможности пакета Mathematica

Построение контурных графиков

Контурные графики, или графики линий равных высот, используются для отображения поверхностей на плоскости. Они удобны для выявления всех экстремумов функций в пределах области графика. Такие графики являются линиями пересечения поверхности с секущими горизонтальными плоскостями, расположенными параллельно друг под другом. Они часто используются в картографии.

Основными функциями и директивами для построения контурных графиков являются следующие:

`ContourPlot[f,{x, xmin, xmax}, {y, ymin, ymax}]` — порождает контурный график f как функции от x и y ;

`ContourGraphics [array]` — представляет контурный график массива `array`;

`ListContourPlot[array]` — формирует контурный график из массива величин высот.

Графические возможности пакета Mathematica

Для управления возможностями графической функции `ContourPlot` используются опции, полный список которых выводит команда `Options [ContourGraphics]`. Помимо уже рассмотренных ранее опций используются следующие:

`ColorFunction` — задает окраску областей между линиями;

`Contours` — задает число контурных линий;

`ContourLines` — задает прорисовку явных (`explicit`) контурных линий;

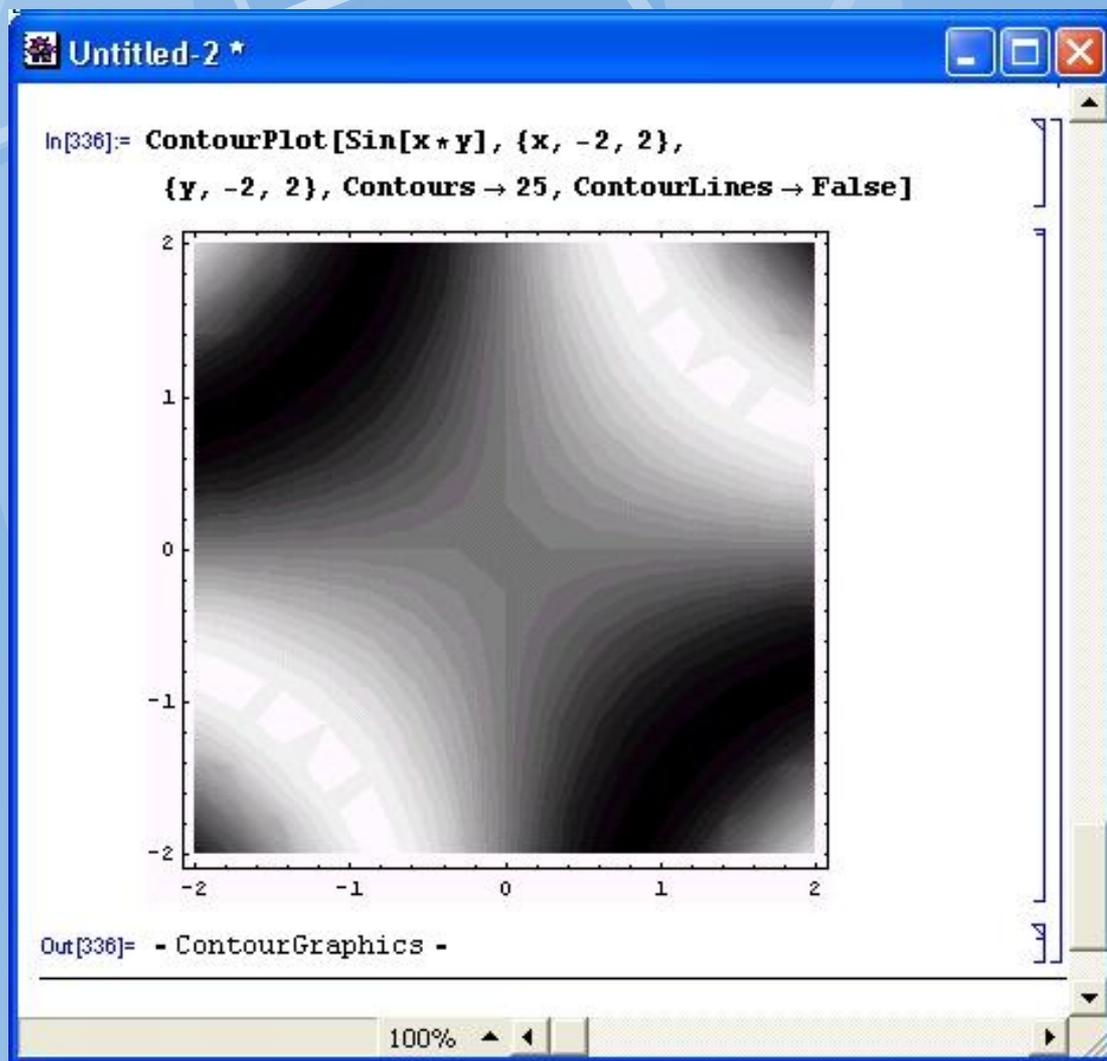
`ContourShading` — задает затенение областей между контурными линиями;

`ContourSmoothing` — задает сглаживание контурных линий;

`ContourStyle` — задает стиль рисуемых линий для контурных графиков;

`MeshRange` — задает области изменения x - и y -координат.

Графические возможности пакета Mathematica



Графические возможности пакета Mathematica

Построение графиков плотности

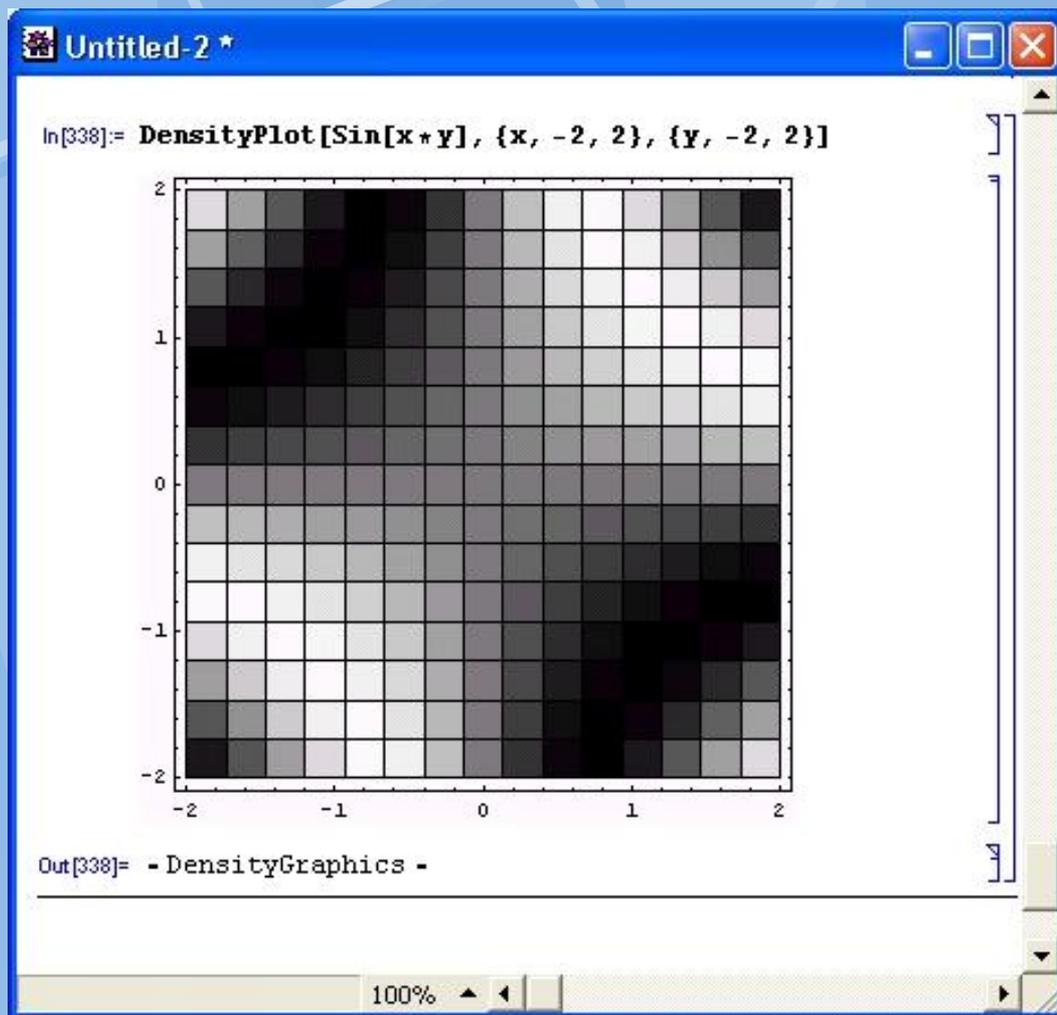
Функцией двух переменных $f(x, y)$ может описываться плотность некоторой среды. Для построения графиков плотности используются следующие графические функции:

`DensityGraphics [array]` — является представлением графика плотности;

`DensityPlot[f, {x, xmin, xmax}, {y, ymin, ymax}]` — строит график плотности f как функции от x и y ;

`ListDensityPlot [array]` — формирует график плотности из массива величин высот.

Графические возможности пакета Mathematica



Лекция окончена

Нажмите клавишу <ESC> для выхода

Графические возможности пакета Mathematica

Графика всегда была козырной картой системы. Графические возможности достигаются как обилием встроенных функций, так и средствами их модификации с помощью директив, опций и примитивов. Система позволяет строить практически любые виды математических графиков, причем обычного пользователя в большинстве случаев удовлетворяют графики, параметры которых система задает по умолчанию.

Начнем рассмотрение графических возможностей системы с построения графиков функций одной переменной вида $y = f(x)$ или просто $f(x)$. График таких функций строится на плоскости, т.е. в двумерном пространстве. Он представляет собой геометрическое место точек (y, x) при изменении независимой переменной x (абциссы) в заданных пределах, например от минимального значения x_{\min} до максимального значения x_{\max} .

Графические возможности пакета Mathematica

Для построения двумерных графиков функций вида $f(x)$ используется встроенная в ядро функция Plot. Она задается в следующих формах:

Plot[f,{x,xmin,xmax}]	- строит график функции f аргумента x в интервале от x_{\min} до x_{\max} .
Plot[{f1,f2,...},{x,xmin,xmax}]	- строит графики ряда функций f_i ($i = 1, 2, \dots$).

Формат построенных графиков задается специальными графическими опциями и директивами.

С функцией Plot используются различные опции, меняющие те или иные параметры графиков и их вид (например, размер графика, наличие осей и т.д.) Опции внутри записей графических функций задаются своим именем name и значением value в виде: name-> value

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

Automatic - используется автоматический выбор;

None - опция не используется;

All - используется в любом случае.

Графические возможности пакета Mathematica

Опции функции Plot

С функцией Plot используются различные опции, меняющие те или иные параметры графиков и их вид (например, размер графика, наличие осей и т.д.). Опции внутри записей графических функций задаются своим именем name и значением value в виде:

name -> value

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

Autonomic – используется автоматический выбор;

None – опция не используется;

All – используется в любом случае.

Графические возможности пакета Mathematica

Ниже представлены основные опции для графических функций (звездочкой отмечены те опции, которые можно использовать и в трехмерной графике):

\$ AspectRatio*	- задает пропорцию размеров графика - отношение высоты к ширине;
Axes*	- устанавливает, должны ли рисоваться оси (по умолчанию Automatic);
AxesLabel*	- устанавливает, нужны ли отметки на осях графиков (по умолчанию None);
AxesOrigin	- указывает, где должно располагаться начало отсчета осей (по умолчанию Automatic);
AxesStyle*	- указывает в каком стиле должны строиться оси графика (по умолчанию Automatic);
Background*	- указывает какого цвета должен быть фон графика (по умолчанию – Automatic);
\$DisplayFunction*	- указывает канал и порядок вывода графического объекта (по умолчанию -\$DisplayFunction);
PlotRange	- указывает какие точки включать в график (по умолчанию Automatic);
PlotLabel*	- указывает нужно ли вывести титульную надпись для графика (по умолчанию None);
\$ PlotStyle	- устанавливает стиль отображения графика (по умолчанию Automatic);
RotateLabel	- указывает, следует ли метки на вертикальных осях развернуть так, чтобы они стали вертикальными;
Frame	- указывает, нужно ли рисовать рамку вокруг графика (по умолчанию False).
Ticks*	- устанавливает штриховые метки для осей (по умолчанию Automatic);

Графические возможности пакета Mathematica

Опции могут серьезно изменить вид графика.

Примеры использования опций графики:

Установка масштаба 2D-графика по оси y

```
Plot[Sin[x]/x,{x,-20,20},PlotRange->{-0.25,1}]
```

Ввод надписей по осям 2D – графика

```
Plot[Sin[x]^3,{x,0,20},AxesLabel->{"x value", "Graphic Sin(x)^3"}]
```

График с титульной надписью

```
Plot[Sin[x]^3,{x,0,20},Axes->None,PlotLabel->"Graphic for functions Sin(x)^3"]
```

Графические возможности пакета Mathematica

Директивы двумерной графики

Еще одним мощным средством графики системы являются графические директивы. Они указывают, с какими графическими параметрами (цвет, толщина линий, их стиль и т.д.) должна строить графики та или иная функция.

В системе используются следующие директивы двумерной графики:

AbsoluteDashing[{d1,d2,...}]	- указывает, что последующие линии должны рисоваться пунктиром со смежными (последовательными) сегментами, имеющими абсолютные длины d1,d2 и повторяющимися циклически (значения длины di задаются в пикселях);
AbsolutePointSize[d]	- устанавливает построение точек графика в виде кругов с абсолютным радиусом d (в пикселях);
AbsoluteThickness[d]	- устанавливает толщину d рисуемых линий;
Dashing[{r1,r2,...}]	- устанавливает вывод линий пунктиром с последовательными сегментами длиной r1, r2,... повторяющимися циклически;
Thickness[r]	- устанавливает толщину r для всех последующих линий r.

Пример использования директив двумерной графики:

```
Plot[Sin[x],{x,0,20},PlotStyle-> Dashing[{0.05,0.025}]]
```

Графические возможности пакета Mathematica

Графическая функция ListPlot

Строить графики часто приходится по точкам. Для этого существует встроенная в ядро системы функция ListPlot.

ListPlot[{y1,y2,...}]	- выводит график списка величин y_1, y_2, \dots и т.д.
ListPlot[{{x1,y1},{x2,y2},...}]	- выводит график списка величин с указанными x_i и y_i координатами ($i=1,2,\dots$).

Отметим опцию для данной функции:

PlotJoined – указывает, следует ли точки, нанесенные на график, соединять отрезками прямых (по умолчанию – True).

Пример использования функции ListPlot:

```
g = ListPlot[{1,2,3,1.5,0.5,0.2},PlotRange->{0,3}]
```

Графические возможности пакета Mathematica

Перестройка и комбинирование графиков

При построении графиков приходится изменять их вид, а также те или иные параметры и опции. Для этого можно повторить вычисления, но тогда заметно снизится скорость работы с системой. Для ее повышения удобно использовать специальные функции перестройки и вывода графиков, учитывающих что узловые точки этих графиков уже были рассчитаны и их координаты хранятся в памяти ПК.

В этом случае удобно использовать следующие функции:

<code>\$ Show[plot]</code>	- построение графика по вычисленным данным;
<code>\$ Show[plot,option->value]</code>	- построение графика с заданной опцией;
<code>\$ Show[plot1,plot2,...]</code>	- построение нескольких графиков с наложением их друг на друга.

Примеры: *Построение объединенного сдвоенного 2D-графика*

```
g1 = Plot[Sin[x],{x,0,20}];
```

```
g2 = Plot[Sin[x]^3,{x,0,20}];
```

```
Show[g1,g2,PlotRange->{-1.5,1.5}]
```

Графические возможности пакета Mathematica

Примитивы двумерной графики

Примитивами двумерной графики называют дополнительные указания, вводимые в функцию `Graphics`, для построения с ее помощью некоторых геометрических фигур. Сама функция `Graphics` задается в следующем виде:

`Graphics[primitives,options]`

Применение примитивов в составе функции `Graphics` избавляет пользователя от задания довольно сложных математических выражений, описывающих эти фигуры, и описания алгоритмов построения по ним графиков фигур. Примитивы могут дополнять и иные функции.

Графические возможности пакета Mathematica

Circle[{x,y},r]	- окружность радиуса r с центром $\{x,y\}$;
Circle[{x,y},{rx,ry}]	- эллипс с полуосями r_x и r_y с центром $\{x,y\}$;
Circle[{x,y},r,{theta1,theta2}]	- дугу окружности радиуса r с углами конечных точек θ_1 и θ_2 с центром $\{x,y\}$;
Disk[{x,y},r]	- закрашенный круг радиусом r с центром $\{x,y\}$;
Disk[{x,y},{rx,ry}]	- закрашенный полуовал с полуосями r_x и r_y с центром $\{x,y\}$;
Disk[{x,y},r,{theta1,theta2}]	- сегмент круга с углами конечных точек образующей окружности θ_1 и θ_2 , радиусом r и с координатами центра $\{x,y\}$;
Line[{{x1,y1},{x2,y2},...}]*	линию в виде отрезков прямых, соединяющих точки с координатами $\{x_1,y_1\}$, $\{x_2,y_2\}$,
Point[{x,y}]*	- точку с координатами $\{x,y\}$.
Text[expr,coords]	- выражение $expr$ в текстовой форме с центром, имеющим координаты $coords$.

Графические возможности пакета Mathematica

Примеры использования примитивов двумерной графики:

Построение четырех графических объектов – линии, окружности, текста и жирной точки.

```
g1 = Graphics[Line[{{-1,-1},{1,1}}]];
g2 = Graphics[Circle[{0,0},0.8]];
g3 = Graphics[Text["Hello!",{0.25,0.5}]];
Show[g1,g2,g3,Graphics[{PointSize[0.032],
Point[{-0.5,0.2},Axes->Automatic]
```

Графические возможности пакета Mathematica

Построение графиков параметрически заданных функций

Функции, графики которых строятся на плоскости, могут быть заданы в параметрическом виде: $x=f_x(t)$ и $y=f_y(t)$, где независимая переменная t меняется от минимального значения t_{\min} до максимального t_{\max} с шагом dt . Особенно удобно применение таких функций для построения замкнутых линий, таких как окружности, эллипсы, циклоиды и др.

Для построения параметрически заданных функций используются следующие графические функции системы:

ParametricPlot {{fx,fy},{t,tmin,tmax}}	- строит параметрический график с координатами f_x и f_y , получаемыми как функции от t ;
ParametricPlot {{fx,fy}, {gx,gy},{t,tmin,tmax}}	- строит графики нескольких параметрических кривых.

Графические возможности пакета Mathematica

Для изменения вида графиков используются директивы и опции, что позволяет реализовать большое разнообразие графиков.

Пример построения графика, заданного параметрически:

```
ParametricPlot[{Sin[2*t],Sin[3*t]},{t,0,2Pi},  
AspectRatio->Automatic]
```

Графические возможности пакета Mathematica

Построение графиков трехмерных поверхностей.

Функция двух переменных $z = f(x,y)$ в пространстве образует некоторую трехмерную поверхность или фигуру. Для их построения приходится использовать координатную систему с тремя осями координат: x , y , z .

Для построения графиков трехмерных поверхностей используются следующие основные графические функции:

ListPlot3D[array]

- строит трехмерный график поверхности, представленной массивом значений высот array;

**Plot3D[f[x,y],{x,xmin,xmax},
{y,ymin,ymax}]**

- строит трехмерный график для аналитически заданной функции f переменных x, y , изменяющихся в заданных пределах;

Графические возможности пакета Mathematica

Для построения трехмерных графиков в параметрической форме также имеются функции:

ParametricPlot3D[fx,fy,fz]	- строит трехмерную поверхность, параметризованную по t и u в заданных пределах их изменения.
ParametricPlot3D[fx,fy,fz}, {t,tmin,tmax}]	- выполняет построение трехмерной кривой, параметризованной по переменной t , которая изменяется от $tmin$ до $tmax$.
и другие.	

Примеры построения трехмерных графиков:

Исходная математическая поверхность

```
g3=plot3D[Cos[x*y],{x,-3,3},{y,-3,3},PlotPoints->40]
```

Математическая поверхность с отсеченной верхней частью

```
Show[g3,PlotRange->{0,0.5}]
```

Графические возможности пакета Mathematica

Примитивы трехмерной графики

Наряду с построением графиков поверхностей, заданных аналитическими выражениями, имеется возможность создания графиков различных объектов с помощью функции

Graphics3D[primitives,options] - представляет трехмерное графическое изображение.

Основные примитивы:

Cuboid[{xmin,ymin,zmin}]

- строит единичный куб с гранями, ориентированными параллельно осям};

**Cuboid[{xmin,ymin,zmin},
{xmax,ymax,zmax},]**

- строит прямоугольный параллелепипед, заданный координатами противоположных вершин.

Графические возможности пакета Mathematica

Примеры:

Построение куба с точками

```
pts=Table[Point[{Random[],Random[],Random[]}],{25}];
```

```
Show[Graphics3D[{PointSize[0.014],pts}]]
```

```
Show[Graphics3D[pts],ViewPoint->{8,2,2}]
```

Построение шести кубов в пространстве

```
g=Graphics3D[{Cuboid[{0,0,0}],Cuboid[{2,2,2}],
```

```
    Cuboid[{1,1,3}],Cuboid[{3,2,1}],
```

```
    Cuboid[{2,1,1}],Cuboid[{3,3,3}]}];
```

```
Show[g]
```

Лекция окончена

Нажмите клавишу <ESC> для выхода