

MATLAB

Установка ПО

МГТУ им. Н.Э. Баумана предоставляет всем сотрудникам и студентам университета (включая все филиалы) лицензию на программные продукты семейств **MATLAB** и **Simulink** (все доступные пакеты).

Для получения лицензии необходимо:

- 1) Открыть браузер и перейти на страницу <http://getmatlab.ru>;
- 2) Нажать кнопку «Запросить лицензию»;
- 3) В открывшейся форме ввести адрес электронной почты.

ВАЖНО: если указанная почта находится в домене @bmstu.ru, лицензия будет выдана в автоматическом режиме в течение нескольких минут, в противном случае лицензия выдается в ручном режиме в течение 1 рабочего дня.

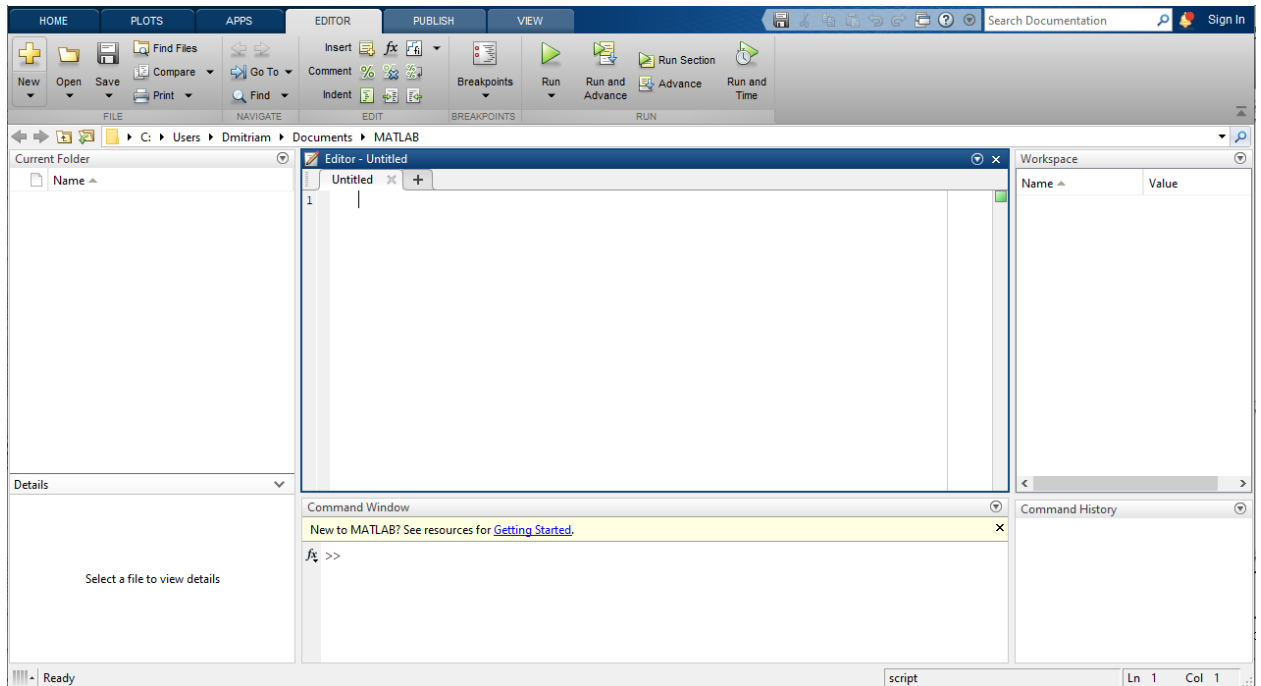
- 4) Заполнить остальные поля формы и нажать кнопку «Отправить запрос».

В течение нескольких минут после отправки запроса на указанный электронный адрес придет письмо с информацией о дальнейших шагах. В случае возникновения трудностей с получением лицензии, вы можете обратиться за технической поддержкой по адресу help@exponenta.ru.

Полезные ссылки

Ссылка	Описание
http://www.mathworks.com	Сайт компании The MathWorks
http://www.mathtools.net	Научный портал, поддерживаемый компанией MathWorks
http://www.exponenta.ru	Образовательный математический сайт
http://sl-matlab.ru	Центр компетенций MathWorks
http://softline.ru	Учебный центр Softline

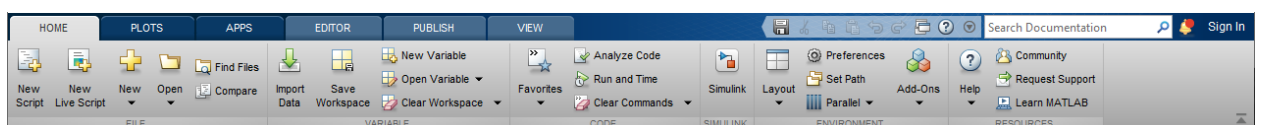
Интерфейс системы Matlab



Интерфейс рабочего стола содержит:

- меню, расположенное в верхней части окна, выделенное серым цветом заливки; содержит вкладки **HOME**, **PLOTS**, **APPS**; начинать работу следует на вкладке **HOME**;
- адресную строку – указатель рабочего каталога, позволяющую выбрать рабочую директорию;
- окно в центре – **Command Window** в котором появляется двойная стрелка (**>>**) указывающая на начало командной строки;
- окно слева – **Current Folder** в котором отображаются файлы текущего каталога и определенные данные описания их свойств;
- окно справа – **Workspace**, представляющие все глобальные переменные, использованные в командах или при выполнении программ;
- окно справа внизу - **Command History**, в котором повторены все команды, выполненные в командной строке.

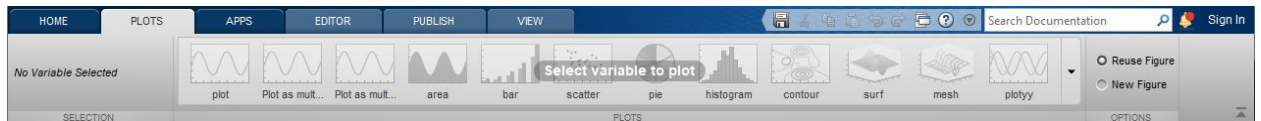
Вкладка HOME



- **New Script** – создание нового файла (.m);
- **New Live Script** – создание интерактивного нового скриптового файла;

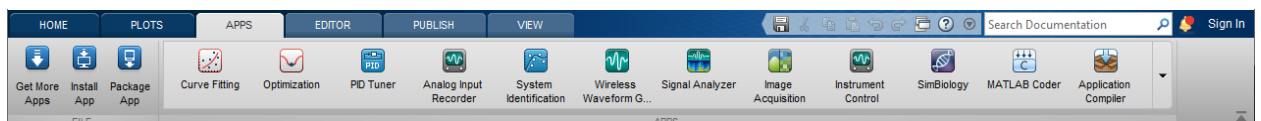
- **Open** – открытие файлов проекта;
- **Compare** – сравнение двух файлов по содержанию;
- **Import Data** – импорт различных данных в Matlab, в том числе и таблиц.
- **Save Workspace** – сохранение глобальных переменных в отдельный файл Matlab;
- **New Variable** – создание новых переменных;
- **Open Variable** – открытие ранее сохраненных переменных;
- **Clear Workspace** – удаление созданных переменных в рабочей сессии;
- Подраздел **CODE** – различные операции с кодом (**Analyze Code** - оптимизация, отладка; **Run and Time** – выполнение кода и измерение времени выполнения; **Clear Commands** - очистка области выполнения команд);
- **Simulink** – запуск модуля Simulink;
- Подраздел **ENVIRONMENT** – настройки программы, настройки расположения окон программы, редактирование рабочих директорий, параллельные вычисления, подключение дополнительных расширений;
- Подраздел **RESOURCES** – Help, полезные материалы Matlab, Community, поддержка.

Вкладка PLOTS

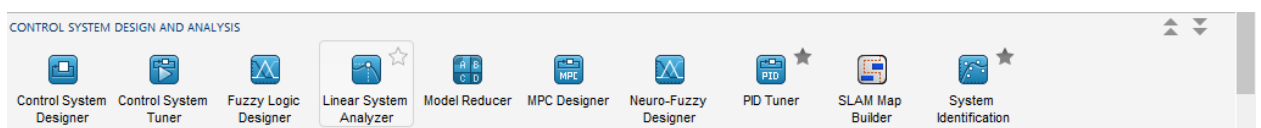


Работа с графиками.

Вкладка APPS



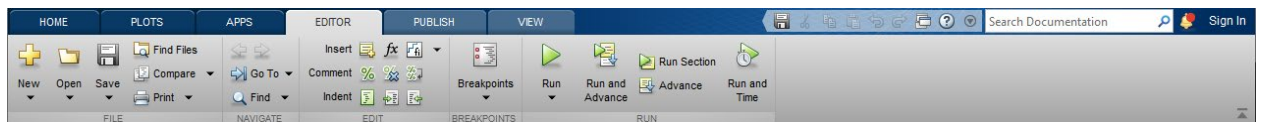
На этой вкладке показаны приложения и модули **Matlab**, которые установлены в системе. Важным элементом для наших проектов является **Control System Design and Analysis**.



Приложение **Linear System Analyzer** – позволяет анализировать временные и частотные характеристики систем LTI. (LTI – линейные стационарные системы, которые описываются линейными дифференциальными уравнениями). Это приложение позволяет:

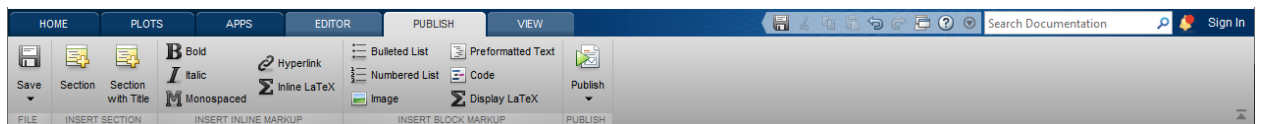
- Просматривать и сравнивать графики отклика систем SISO и MIMO или нескольких линейных моделей одновременно;
- Генерировать графики отклика системы на входные воздействия;
- Создавать графики частотных характеристик системы;
- Просматривать основные характеристики отклика системы на входные воздействия.

Вкладка EDITOR



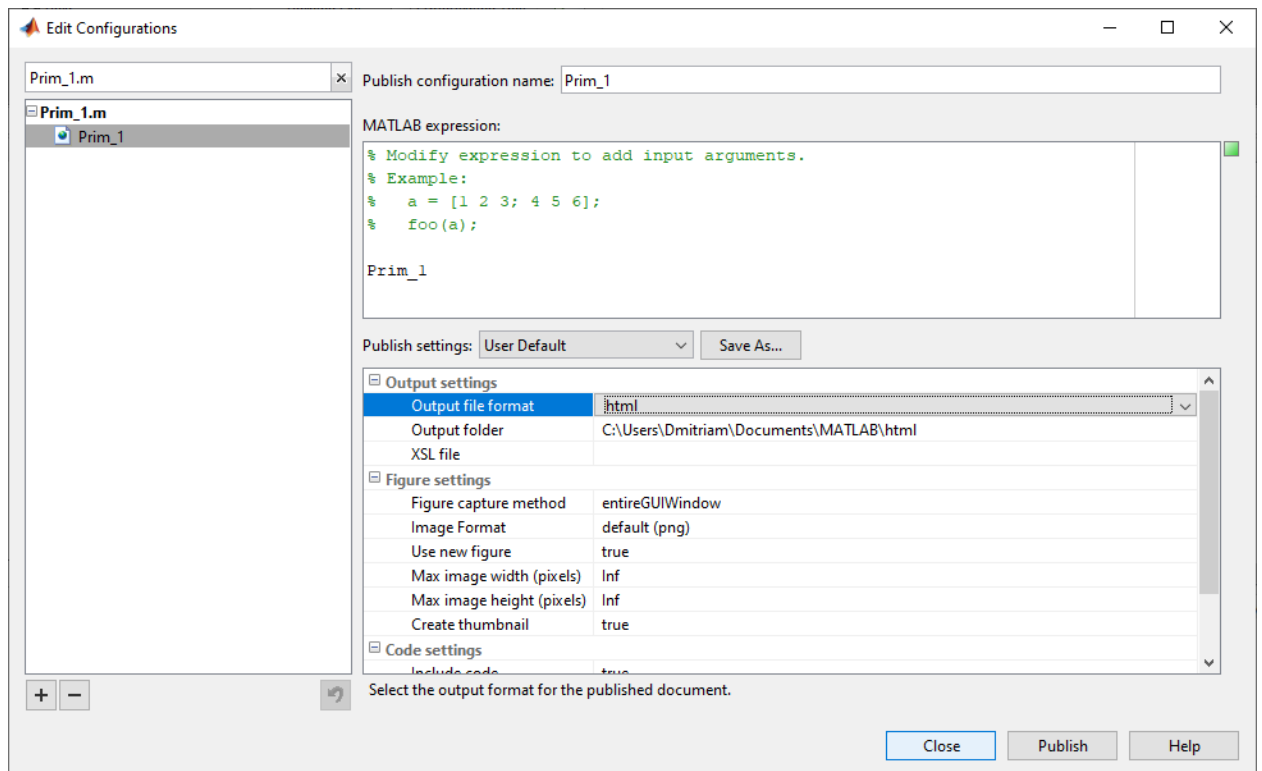
Появляется при создании или открытии файла программы (скрипта) Matlab. Скрипт – текстовый файл с расширением «.m». В данной вкладке доступны различные инструменты для работы с текстовыми файлами Matlab (создание, открытие, сохранение, сравнение файлов; добавление секций, встроенных функций, комментариев в код программы; отладка и различные режимы исполнения сценариев).

Вкладка PUBLISH

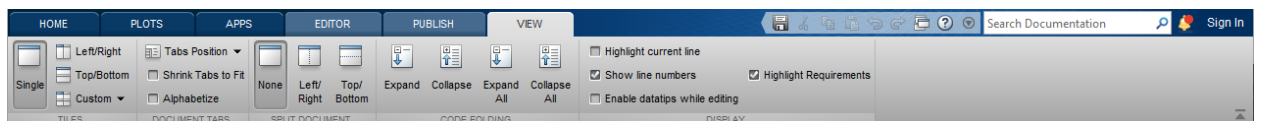


Позволяет делать вывод программы со всеми результатами в читаемый и редактируемый файл. Результаты выводятся по секциям. Позволяет создавать отчет в форматах – html, xml, doc, ppt, pdf.

Настройка формирования отчета по написанной программе вызывается нажатием кнопки «Publish» → «Edit Configurations».



Вкладка VIEW



Настройка внешнего вида рабочего пространства.

Основы программирования MATLAB

Matlab – интерпретатор и система программирования. Позволяет вводить операторы по одному в командную строку в интерактивном режиме или запускать m-файлы, в которых операторы заданы в виде списка.

Математические выражения

В состав математических выражений входят: переменные, числа, операторы, функции.

Переменная – это массив (матрица), вектор или скаляр. Matlab не требует какого-либо описания типа переменной или размерности массива.

Имена переменных, констант и функций могут быть составлены из любых символов латинского алфавита, цифр, знака подчеркивания, но должно начинаться с буквы. Для идентификации переменной используются первые 31 символ имени.

Системные константы

pi = 3.14159265358979;

i = мнимая единица, то же **j**;

Inf = бесконечность, результат деления на 0;

NaN = неопределенное значение;

eps = 2^{-52} или $2.2204e^{-16}$ – характеризует точность вычислений с плавающей точкой.

Операторы Matlab

Язык Matlab – это язык операторов. Операторы задаются по одному в командной строке для исполнения в интерактивном режиме или в виде списка в m-файле или в script-файле. Фактически m-файл является программой, которая интерпретируется и выполняется системой Matlab. После выполнения m-файла в операционной среде системы, до завершения сеанса, остаются все значения глобальных переменных, и они доступны для выполнения любых действий путем задания операторов в командной строке.

Формы записи операторов:

- с явным присвоением: *переменная = выражение*;
- с неявным присвоением: *выражение*;

Оператор содержит:

- имена переменных и числовые константы;
- имена функций;
- специальные символы указывающие на выполняемые действия $+$ $-$ $*$ $/$ $^$ и на порядок действий $()$;
- символ $(;)$ указывающий на завершение строки матрицы и на подавление вывода результата на экран;
- разделители операторов при записи более одного оператора в строке $(,)$;
- символы продолжения строки для записи более 256 символов – точки, не менее двух;
- пробелы, в любых местах, они не влияют на выполняемые действия и служат для оформления строки;
- символ $\%$ указывает на то, что следующие за ним символы являются комментарием, с него можно начать строку.

Синтаксис операторов:

- В MATLAB используются все буквы латинского алфавита от A до Z, цифры от 0 до 9;

- Большие и малые буквы – различаются системой.

Операторы:

- 1) Арифметические операторы: + - * /
- 2) Логические операторы:

Логическое И	&; and (and(a, b))
Логическое ИЛИ	; or (or(a, b))
Логическое НЕ	~; not (not(a, b))
Исключающее ИЛИ	xor (xor(a, b))
Верно, если все элементы вектора равны нулю	any (any(a))
Верно, если все элементы вектора не равны нулю	all (all(a))

- 3) Операторы отношения:

Равно	==; eq (eq(a, b))
Не равно	~=; ne (ne(a, b))
Больше	>; gt (gt(a, b))
Больше или равно	>=; ge (ge(a, b))
Меньше	<; lt (lt(a, b))
Меньше или равно	<=; le (le(a, b))

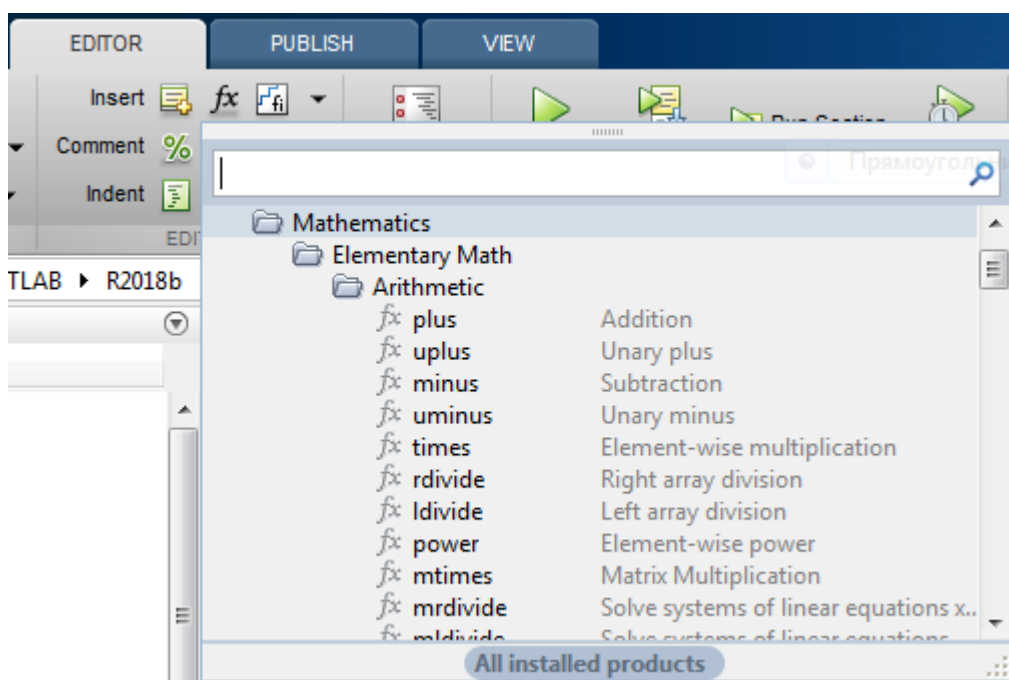
Для арифметических операторов есть эквивалентные функции:

Сложение	+	plus (plus(A,B))
Унарный плюс	+	uplus (uplus(A))
Вычитание	-	minus (minus(A,B))
Унарный минус	-	uminus (uminus(A))
Поэлементное умножение	.*	times (times(A,B))
Умножение матриц	*	mtimes (mtimes(A,B))
Поэлементное правое деление	./	rdivide (rdivide(A,B))
Матричное правое деление	/	mrdivide (mrdivide(A,B))
Поэлементное левое деление	.\	ldivide (ldivide(A,B))
Матричное левое деление	\	mldivide (mldivide(A,B))
Поэлементная степень	.^	power (power(A,B))

Матричная степень	\wedge	mpower (mpower(A,B))
Транспонирование	\cdot'	transpose (transpose(A))
Комплексное сопряженное транспонирование	\cdot'	ctranspose (ctranspose(A))

Дополнительную информацию по каждому оператору можно посмотреть с помощью команды «help ops»

Любой оператор можно добавить в текст программы нажатием кнопки «fx» во вкладке EDITOR. Или в командной строке также кнопкой «fx».



Форматы чисел

Диапазон представления чисел при вычислениях $10^{-308} - 10^{308}$, все внутренние вычисления производятся с двойной точностью.

Запись действительных чисел выполняется:

- в десятичной форме, знак плюс и точка у целых чисел не обязательны;
- в показательной форме по основанию десять; показатель степени отделен от мантиссы символом e или E, пробел не допускается;
- комплексное число представлено действительной и мнимой частями, при мнимой части проставлен символ i или j (без знака умножения).

Основные форматы чисел можно посмотреть с помощью ввода команды «help format»:

```
>> help format
```

```
format          Set output format.
```


format SHORT	Scaled fixed point format with 5 digits.
format LONG	Scaled fixed point format with 15 digits for double and 7 digits for single.
format SHORTE	Floating point format with 5 digits.
format LONGE	Floating point format with 15 digits for double and 7 digits for single.
format SHORTG	Best of fixed or floating point format with 5 digits.
format LONGG	Best of fixed or floating point format with 15 digits for double and 7 digits for single.
format SHORTENG	Engineering format that has at least 5 digits and a power that is a multiple of three
format LONGENG	Engineering format that has exactly 16 significant digits and a power that is a multiple of three.

Запись операторов в командном окне Matlab

При записи оператора с неявным присвоением, результат вычислений присваивается автоматически внутренней переменной **ans** (**answer**), может быть вызван и использован по этому имени и сохраняется до выполнения следующего оператора с неявным присвоением.

Примеры записи переменных, скаляров и матриц, выполнение простых операторов:

```
>> a = 2; b = 5;           % символ ; подавляет эхо-вывод
>> c = (a + b) * 2
c =
    14
>> pi                      % встроенная константа
ans =
    3.1416
>> i                      % комплексные числа, i - встроенная константа
ans =
    0.0000 + 1.0000i
```

Элементарные функции Matlab

Matlab имеет встроенные элементарные математические функции. Информацию о них можно узнать с помощью команды «**help elfun**».

Модуль	abs(x)
--------	--------

Экспонента	exp(x)
Натуральный логарифм	log(x)
Логарифм по основанию 2	log2(x)
Десятичный логарифм	log10(x)
2 в степени x	pow(x)
Квадратный корень	sqrt(x)

Все встроенные элементарные функции должны записываться в программах **малыми буквами!**

Так же есть специальные функции, сгруппированные по разделам:

- help elmat – Elementary matrices and matrix manipulation (Zeros, Ones, Size)
- help specfun – Specialized math functions (Factorial, Coordinate transforms)

Векторы и матрицы

Ввод матриц может осуществляться двумя способами:

- a = [1 2 3; 4 5 6; 7 8 9];
- a = [1 2 3;
4 5 6;
7 8 9];

Операции с матрицами:

Сложение матриц

```
>> a = [1 2; 3 4];
>> b = [5 6; 7 8];
>> c = a + b                                % с присвоением переменной
c =
     6     8
    10    12
>> a + b                                    % с неявным присвоением
ans =
     6     8
    10    12
```

Задание одного элемента матрицы

```
>> a = [1 2 3; 4 5 6; 7 8 9];
>> a(1,1)
ans =
     1
```

```
>> a(2,3)
ans =
    6
```

Транспонирование

```
>> a'
ans =
    1    4    7
    2    5    8
    3    6    9
```

Присоединение столбца

```
>> a = [1 2; 3 4]; b = [5; 6];
>> s1 = [a, b]
s1 =
    1    2    5
    3    4    6
```

Присоединение строки

```
>> a = [1 2; 3 4]; c = [5 6];
>> s2 = [a;c]
s2 =
    1    2
    3    4
    5    6
```

Сложение и вычитание вектора-строки и вектора-столбца или векторов разных размеров приводит к ошибке. Операция `*` предназначена для умножения векторов по правилу матричного умножения. Поскольку MatLab различает вектора-строки и вектора-столбцы, то допустимо либо умножение вектора-строки на такой же по длине вектор-столбец (скалярное произведение), либо умножение вектора-столбца на вектор-строку (внешнее произведение, в результате которого получается прямоугольная матрица).

Функции обработки векторов:

Функции	Назначение
<code>s = sum(a)</code>	Сумма всех элементов вектора a
<code>p = prod(a)</code>	Произведение всех элементов вектора a
<code>m = max(a)</code>	Нахождение максимального значения среди элементов вектора a
<code>[m,k] = max(a)</code>	Второй выходной аргумент k содержит номер максимального элемента в векторе a
<code>m = min(a)</code>	Нахождение минимального значения среди элементов вектора a

<code>[m,k] = min(a)</code>	Второй входной аргумент k содержит номер минимального элемента в векторе a
<code>m = mean(a)</code>	Вычисление среднего арифметического элементов вектора a
<code>a1 = sort(a)</code>	Упорядочение элементов вектора по возрастанию
<code>[a1,ind] = sort(a)</code>	Второй выходной аргумент ind является вектором из целых чисел от 1 до length(a) , который соответствует проделанным перестановкам
<code>L = length(a)</code>	Нахождение длины вектора
<code>x1 = fliplr(x)</code>	Переворот вектора (матрицы) слева направо
<code>y1 = rot90(y)</code>	Поворот матрицы на 90 градусов против часовой стрелки
<code>S = dot(a,b)</code>	Скалярное произведение двух векторов
<code>C = cross(a,b)</code>	Векторное произведение определено только для векторов из трех элементов

Вызвав справку по приведенным в таблице функциям (**help funname**) можно найти много других функций, предназначенных для преобразования векторов и матриц.

Для операции транспонирования зарезервирован апостроф **'**. Если вектор содержит комплексные числа, то операция **'** приводит к комплексно-сопряженному вектору. При вычислении скалярного и векторного произведений функциями **cross** и **dot** не обязательно следить за тем, чтобы оба вектора были либо столбцами, либо строками. Результат получается верный, например, при обращении **c=cross(a,b')**, только **c** становится вектором-строкой.

Для обработки матриц также существуют специальные функции, например, функции для создания стандартных матриц: **zeros**, **eye**, **ones**, **rand**, **diag** (см. **help matlab\elmat**).

Примеры использования оператора **(:)**:

Использование оператора (:)	
<pre>>> a = [1 2 3 4 5 6 7 8]; >> sum(a(:,1))</pre>	% знак (:) → операция со столбцом
<pre>ans = 6</pre>	
<pre>>> a(2,:) ans = 5 6 7 8</pre>	% операция со строкой
<pre>>> 1:6</pre>	% интервал значений целых чисел

<pre>ans = 1 2 3 4 5 6 >> a = 1:6 % с присвоением a = 1 2 3 4 5 6</pre>
<pre>>> a = 1.1:5.5 % с десятичными знаками, шаг единица a = 1.1000 2.1000 3.1000 4.1000 5.1000</pre>
<pre>>> a = 0.1:0.1:0.5 % дробный шаг a = 0.1000 0.2000 0.3000 0.4000 0.5000</pre>
<p>Индексация двоеточием позволяет выделить идущие подряд элементы в новый вектор. Начальный и конечный номера указываются в круглых скобках через двоеточие, например:</p> <pre>>> z = [0.2 -3.8 7.9 4.5 7.2 -8.1 3.4]; >> znew = z(3:6) znew = 7.9000 4.5000 7.2000 -8.1000</pre>
<p>Вызов функции prod с заданием интервала с помощью двоеточия вычисляет произведение элементов вектора z со второго по шестой:</p> <pre>>> p = prod(z(2:6)) p = 7.8784e+03</pre>
<p>Указание номеров элементов вектора можно использовать и при вводе векторов, последовательно добавляя новые элементы (не обязательно в порядке возрастания их номеров).</p> <p>Команды:</p> <pre>>> h = 10; h(2) = 20; h(4) = 40;</pre> <p>приводят к образованию вектора:</p> <pre>>> h h = 10 20 0 40</pre>
<p>Примечание: Для ввода первого элемента h не обязательно указывать его индекс, т.к. при выполнении оператора h=1 создается вектор (массив размера один на один). Следующие операторы присваивания приводят к автоматическому увеличению длины вектора h, а пропущенные элементы (в нашем случае h(3)) получают значение ноль.</p>
<p>Индексация вектором служит для выделения элементов с заданными индексами в новый вектор. Индексный вектор должен содержать номера требуемых элементов, например:</p> <pre>>> z = [0.2 -3.8 7.9 4.5 7.2 -8.1 3.4]; >> ind = [3 5 7]; >> znew = z(ind) znew = 7.9000 7.2000 3.4000</pre>

Вектора-столбцы с одинаковым числом элементов можно складывать и вычитать друг из друга при помощи знаков "+" и "-". Такое действие верно и для векторов-строк:

```
>> a = [1; 2; 3; 4];
```

```
>> b = [5; 6; 7; 8];
```

```
>> c = a+b
```

```
c =
```

```
6
```

```
8
```

```
10
```

```
12
```

```
>> d = b-a
```

```
d =
```

```
4
```

```
4
```

```
4
```

```
4
```

Особенности оператора умножения (деления, возведения в степень)

MatLab поддерживает два вида вычислительных операций с векторами и матрицами: матричные, выполняемые по правилам линейной алгебры и табличные, выполняемые поэлементно. Знак-точка (.) отличает табличные операции от матричных. Так, наряду с умножением по правилу матричного умножения, существует операция поэлементного умножения .* (точка со звездочкой). Данная операция применяется к векторам одинаковой длины и приводит к вектору той же длины, что исходные, элементы которого равны произведениям соответствующих элементов исходных векторов. Аналогично может быть выполнена операция с матрицами одинаковой размерности, в этом случае матрицы - операнды обрабатываются системой как таблицы, выполняется перемножение соответствующих элементов таблиц, результатом будет матрица такой же размерности. Например, для введенных ранее матриц **a** и **b**:

Операция с матрицами

```
>> a = [1 2; 3 4];
```

```
>> b = [5 6; 7 8];
```

```
>> a * b
```

```
ans =
```

```
19 22
```

```
43 50
```

Операция с массивами таблицами

```
>> a.*b
```

```
ans =
```

```
5 12
```

```
21 32
```

Аналогичным образом выполняется поэлементное деление `./` (точка с косой чертой). Кроме того, операция `.\` (точка с обратной косой чертой) осуществляет обратное поэлементное деление, то есть выражения `a./b` и `b.\a` эквивалентны. Возведение элементов вектора `a` в степени, равные соответствующим элементам `b`, производится с использованием `.^`. Для транспонирования векторов-строк или векторов-столбцов предназначено сочетание `.'` (точка с апострофом). Операции `'` и `.'` для вещественных векторов приводят к одинаковым результатам. Не требуется применять поэлементные операции при умножении вектора на число и числа на вектор, делении вектора на число, сложении и вычитании вектора и числа. При выполнении, например, операции `a*2`, результат представляет собой вектор того же размера, что и `a`, с удвоенными элементами.

Аппроксимация в Matlab

Аппроксимация данных производится с помощью функции `polyfit(x,y,n)`, которая находит коэффициенты полинома `p(x)` степени `n` методом наименьших квадратов. Для определения значений полинома в заданных `x` применяется функция `polyval`. Представить полученные результаты в графическом виде можно с помощью функции `plot()`.

Аппроксимация данных полиномом и построение графиков

```
x = 1:10;
y = [0 10 0 -10 0 10 0 -10 0 10];
x1 = 1:0.1:10;
n = 9;
p = polyfit(x,y,n);
z = polyval(p,x1);
% Построение графиков
figure('Name','График');
plot(x,y,'b',x1,z,'o-r');
hold on;
title('Аппроксимация');
grid on;
xlabel('x');
ylabel('y');
legend('y','y approx')
hold off;
```

Справочная система (HELP)

Для обращения к справочной системе необходимо в командном окне MATLAB набрать команду:

» **help**

При этом будет представлен перечень разделов (HELP topics) справочной системы. Ниже приведены разделы, на которые следует обратить внимание в первую очередь.

» **help matlab\ops** – выводит перечень операторов и специальных символов, используемых в системе.

» **help arith** – об арифметических операторах,

» **help punct** – об использовании специальных символов в командах,

» **help colon** – о применении специального символа **:** (двоеточие), который управляет выполнением ряда важных операций с матрицами.

» **help matlab\lang** – описание языка системы для работы в режиме интерпретации команд и программирования (написания М- файлов).

» **help matlab\elmat** – простые матрицы и базовые операции с матрицами.

» **help matlab\elfun** – элементарные, базовые функции системы, в том числе тригонометрические, экспоненциальные, обработки комплексных чисел и т.д.

» **help matlab\matfun** – функции линейной алгебры и матричного анализа.

» **help matlab\polyfun** – функции работы с полиномами и интерполяции.

» **help matlab\plotxy** – построение графиков по двум координатным осям.

Полезные команды и функции

Полезные команды и функции

>> **clear** – очистка **Workspace**

>> **clear var** – очистка переменной **var**

>> **clear globals** – очистка глобальных переменных

>> **clc** – очистка командного окна

>> **home** – возврат курсора в ВЛУ

>> **clf reset** – очистка окна графика

>> **who** – просмотр переменных рабочего пространства в процессе решения задач