



# MathWorks MATLAB & Simulink

Основы программирования в MathWorks MATLAB

Андрей Масленников

Москва, 2017

# Тема лекций: Основы программирования в MathWorks MATLAB

1. MathWorks MATLAB & Simulink.
2. Основы синтаксиса.
3. Построение и настройка графиков.
4. Функции.
5. Литература.

# **MathWorks MATLAB & Simulink**

# MathWorks MATLAB & Simulink

## MathWorks MATLAB

Программный комплекс для научных вычислений. Содержит огромное количество функций, необходимых для инженерных и научных расчетов.

## MathWorks Simulink

Программная среда для математического моделирования при моделях, заданных в виде различных схем.

Сайт:

[mathworks.com](https://www.mathworks.com)

Документация:

[mathworks.com/help/](https://www.mathworks.com/help/)

# MathWorks MATLAB & Simulink

## Получение университетской лицензии

### Алгоритм получения лицензии и скачивание дистрибутива:

1. идем на [matlab.ru/education/mgtu-installation](http://matlab.ru/education/mgtu-installation); *(делаем то, что там написано)*
2. идем на [mathworks.com/login](http://mathworks.com/login) и делаем аккаунт;  
*(желательно, на почту в домене @student.bmstu.ru)*
3. заходим в созданный аккаунт, ищем “Associate License”;  
*(Вводим код активации полученный на email из п. 1)*
4. ищем “My Account” и видим следующую строчку:

License	Label	Option	Use
906991	Campus	Total Headcount	Academic

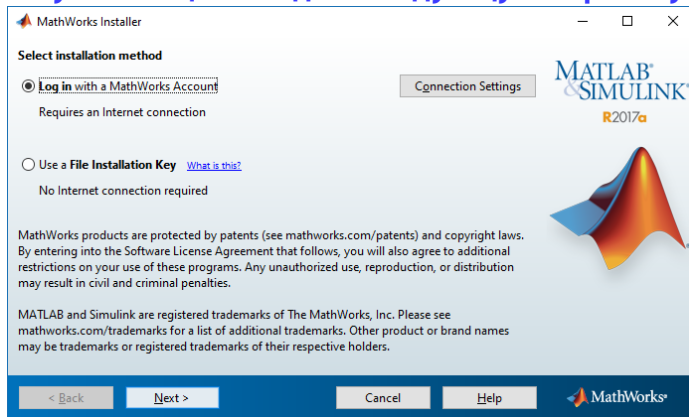
*(жмем стрелочку вниз и скачиваем установщик  $\approx 180$  МБ)*

MathWorks MATLAB с недавних пор работает только на 64-битных ОС

# MathWorks MATLAB & Simulink

## Установка

После запуска установщика видим следующую картинку:



Выбираем “Log in with MathWorks Account”, вводим логин/пароль  
Далее в процесс установки не вмешиваемся  
Установщик будет скачивать  $\approx 7$  ГБ данных

**Окно MATLAB может состоять из нескольких функциональных окон:**

- **Command Window;**  
*(командная строка - место ввода команд и вывода результатов)*
- **Editor;**  
*(окно редактирования скриптов)*
- **Workspace;**  
*(отображает текущие переменные в памяти MATLAB'a)*
- **Current Folder;**  
*(отображает файлы в текущей активной директории)*
- **Figures.**  
*(окно или окна с построенными графиками)*

**Настроить внешний вид можно через Drag & Drop и меню Layout**

### Механизмы реализации вычислений:

- через командную строку;  
*(команда/команды вводятся “ручками” и нажимается Enter)*
- через скрипт;  
*(программный код пишется в редакторе (Editor), после чего сохраняется в .m файл и запускается на выполнение нажатием кнопки Run или F5 или CTRL+Enter (также работает для отдельной секции))*
- через живой скрипт (Live Scripts).  
*(программный код пишется также в редакторе, открытым в специальном режиме, а результаты отображаются не в командной строке, а в редакторе сбоку)*

Использование живых скриптов удобно для презентаций



### Типы файлов используемые в MathWorks MATLAB:

- `.m`;  
*(скрипт или функция)*
- `.mlx`;  
*(живой скрипт)*
- `.mat`;  
*(для хранения данных)*
- `.mdl` и `.slx`;  
*(модель в Simulink, лучше использовать .slx)*
- и др.

# Основы синтаксиса

# Инициализация скрипта

## Очистка командной строки:

```
clc;
```

## Очистка памяти:

```
clear variables;  
clear all;
```

## Заккрытие окон, соединений, открытых файлов и др.:

```
close all;  
close all force;
```

Выполнение скрипта осуществляется нажатием Run или F5

### Скрипт:

```
%% Calculate and Plot Sine Wave
```

```
x = 0:1:6*pi;
```

```
y = sin(x);
```

```
plot(x,y)
```

```
%% Modify Plot Properties
```

```
title('Sine Wave')
```

```
xlabel('x')
```

```
ylabel('sin(x)')
```

```
fig = gcf;
```

```
fig.MenuBar = 'none';
```

Выполнение отдельной секции осуществляется CTRL+ENTER

# Типы данных в MathWorks MATLAB

MATLAB не является языком с жесткой типизацией

## Основные типы данных:

- вещественные (single / double);
- $n$ -мерные числовые массивы;
- строки (string);
- массивы данных (cell);
- структуры;
- указатели на функции.

По умолчанию, все числовые значения хранятся как double

## Data Types:

[mathworks.com/help/matlab/data-types\\_data-types.html](https://mathworks.com/help/matlab/data-types_data-types.html)

# Инициализация скалярных переменных

Присвоение значения переменной с выводом:

```
a = 5
```

Присвоение значения переменной с подавлением вывода:

```
a = 5;
```

Вывод значения переменной:

```
a  
    % alternative  
disp(a);  
    % alternative  
fprintf('a = %2.2f\na = %02d', a, a);
```

Спецификаторы формата ввода/вывода см. в документации по `sprintf`

# Инициализация векторных переменных

## Инициализация вектора-строки:

```
a = [1 2 3];  
b = [1, 2, 3];           % alternative
```

## Инициализация вектора-столбца:

```
a = [1; 2; 3];  
b = [1, 2, 3]';         % alternative
```

## Задание вектора значений с известным шагом:

```
t = (0 : 0.1 : 5)';
```

## Вектор столбец из пяти нулей:

```
a = zeros(5, 1)
```

## Вектор строка из пяти единиц:

```
a = ones(1, 5);
```

## Работа с элементами вектора

**Изменение одного элемента вектора:**

```
a(2) = 5;
```

**Изменение нескольких элементов вектора:**

```
a(1:2) = [0.9; 0.7];
```

**Удаление элементов из вектора:**

```
a(1:2) = [];
```

**Умножение вектора на константу:**

```
b = [4; 2; 1] * 0.5;
```

**Поэлементное возведение в степень:**

```
b = [4; 2; 1] .^ 2;
```



# Объединение векторов

## Вертикальная конкатенация векторов

```
a = [1; 2; 3];  
z = [0.9; 0.7];  
b = vertcat(a, z);  
c = [a; z];           % alternative
```

## Горизонтальная конкатенация векторов

```
a = [1, 2, 3];  
z = [0.9, 0.7];  
b = horzcat(a, z);  
c = [a, z];           % alternative
```

## Генерация случайных чисел

**Вектор нормально распределенных случайных чисел:**

```
a = randn(3, 1);
```

**Вектор равномерно распределенных случайных чисел:**

```
a = rand(3, 1);
```

**Вектор равномерно распределенных случайных целых чисел [1...5]:**

```
a = randi(5, 3, 1);
```

**Белый шум мощностью  $p$  децибел-ватт:**

```
a = wgn(100, 1, p);
```

**Добавление белого шума к сигналу  $y$  с Signal-to-Noise Ratio  $p$ :**

```
y = awgn(y, p);
```

Существуют и более сложные механизмы для более сложных задач

# Матрицы и операции над ними

**Зададим две матрицы  $3 \times 3$ :**

```
A = randn(3);  
B = randn(3, 3);
```

**Умножение матриц:**

```
C = A * B;
```

**Поэлементное умножение:**

```
C = A .* B;
```

**Создание единичной матрицы:**

```
I = eye(3);
```

**Варианты нахождения обратной матрицы:**

```
Inverse = inv(C);  
Inverse = I / C;  
Inverse = C^(-1);    % best option
```

# Матрицы и операции над ними

## Добавление к матрице столбца:

```
C = horzcat(C, [7; 7; 7]);  
C = [C [7; 7; 7]]; % alternative
```

## Добавление к матрице строки:

```
C = vertcat(C, [7 7 7]);  
C = [C; [7 7 7]]; % alternative
```

## Обращение к подматрице:

```
C = C(1:3, 1:3);
```

## Обращение к всем элементам 2 столбца:

```
R = C(:, 2);
```

## Обращение к всем элементам 2 строки:

```
R = C(2, :);
```

# Матрицы и операции над ними

**Минимальное значение по столбцам:**

```
Mn = min(C);
```

**Минимальное значение по всей матрице:**

```
Mn = min(min(C));
```

**Евклидова норма:**

```
Nrm = norm(C, 2);
```

**Удаление строки из матрицы:**

```
C(2, :) = [];
```

**Удаление столбца из матрицы:**

```
C(:, 2) = [];
```

## Логическая индексация:

```
t = (0.0 : 0.01 : 20.0)';  
u = zeros(size(t));  
u(t >= 2.0) = 1.0;
```

## Определение длины вектора:

```
x = length(t);
```

## Определение объема данных в переменной:

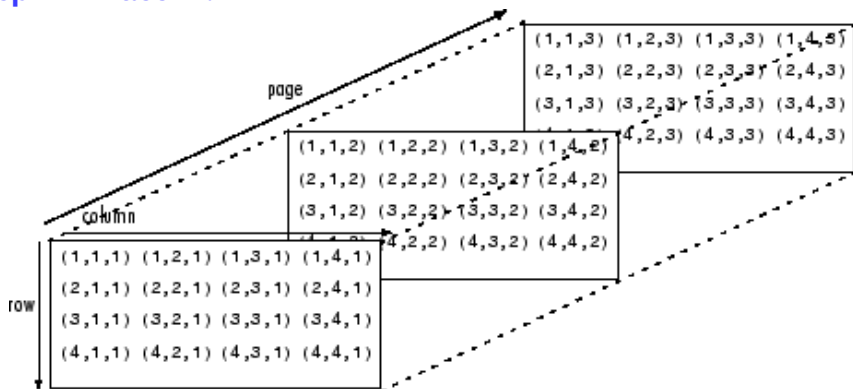
```
x = size(t);
```

## Определение объема данных переменной в $i$ -ой размерности:

```
x = size(t, i);
```

# Многомерные массивы

## Трёхмерный массив:



# Многомерные массивы

**Создание простой матрицы - страница 1:**

```
A = [5 7 8; 0 1 9; 4 3 6];
```

**Расширение матрицы A до многомерного массива:**

```
A(:, :, 2) = [1 0 4; 3 5 6; 9 8 7];
```

**Получить ВСЕ элементы многомерного массива в виде вектора:**

```
R = A(:);
```

**Удалим одну страницу. A станет простой матрицей:**

```
A(:, :, 1) = [];
```

**Удаление единичных размерностей:**

```
A = squeeze(A);
```



## Цикл с известным числом повторений:

```
for i = 1 : 1 : 5  
    disp(i);  
end
```

## Цикл с предусловием:

```
k = 5;  
eps = 1;  
while k > eps  
    disp(k);  
    k = k - 1;  
end
```

## Условный оператор:

```
a = 5;  
if a > 5  
    disp( 'BOLSHE' )  
elseif a == 5  
    disp( 'RAVNO' );  
else  
    disp( 'MENSHE' );  
end
```

## Оператор множественного выбора:

```
a = 2;  
switch a  
    case 1  
        disp( '1' );  
    case 2  
        disp( '2' );  
    otherwise  
        disp( 'WRONG CHOICE' );  
end
```

В отличие от языков низкого уровня в значениях case могут быть не только целые числа, но и символы и целые строки

# Переменные строкового типа

## Создание строки:

```
s = 'some text string';
```

## Строка - массив, элементы - символы:

```
a = s(1:4);
```

## Замена строки строкой в строке:

```
s = strrep(s, 'some ', 'awesome');
```

## Конкатенация строк (пробелы не учитываются):

```
b = strcat(a, ' ', s);
```

## Конкатенация через форматирование:

```
b = sprintf('%s %s', a, s);
```

# Массивы данных

Могут хранить информацию любых типов одновременно!

**Создание массива данных в виде вектора:**

```
C = cell(4, 1);
```

**Пример заполнения:**

```
C{1} = randn(5);
```

```
C{2} = 'string';
```

```
C{3} = 5;
```

```
C{4} = randn(5, 1);
```

**Работа с массивом данных:**

```
T = C(2:3);
```

**Работа с элементами массива данных:**

```
Y = C{2:3};
```

Комментируется весь код за символом %:

```
a = 5; % comment  
a % = 10;
```

Полезные горячие клавиши:

- CTRL+R - закомментировать выделенные строки кода;
- CTRL+T - раскомментировать выделенные строки кода.

## **Построение и настройка графиков**

# Построение графиков

## Создаем сигналы:

```
t = (0.0 : 0.01 : 2.0)';  
s1 = sin(2*pi*t);  
s2 = cos(2*pi*t);
```

## Рисуем две кривых на одном графике:

```
figure(1);  
clf;  
plot(t, s1);  
hold on;  
plot(t, s2);  
hold off;  
grid on;  
grid minor;
```



## Построение графиков

### Два горизонтально расположенных графика:

```
figure(1);  
clf;  
subplot(1, 2, 1)  
plot(t, s1);  
subplot(1, 2, 2)  
plot(t, s2);
```

### Два вертикально расположенных графика:

```
figure(2);  
clf;  
subplot(2, 1, 1)  
plot(t, s1);  
subplot(2, 1, 2)  
plot(t, s2);
```

# Построение графиков

## Базовые настройки:

```
H = figure(1);  
clf;  
plot(t, s1, '-', 'LineWidth', 2.0);  
hold on;  
plot(t, s2, 'x', 'MarkerSize', 5.0);  
hold off;  
grid on;  
grid minor;  
title('Sin/Cos Function Plot')  
xlabel('Time, [s]')  
ylabel('Amplitude')
```

## Сохранение в файле:

```
print(H, '-dpng', '-r150', 'file.png');
```

# Построение графиков

## Настройка через gca:

```
ax = gca;  
ax.FontSize          = 16;  
ax.Title.String      = 'Sin/Cos Function Plot';  
ax.Title.FontSize    = 20;  
ax.Title.FontName     = 'Times New Roman Cyr';  
ax.Title.Color       = 'c';  
ax.XLabel.String     = 'Time, [s]';  
ax.XLabel.FontSize   = 16;  
ax.XLabel.FontName   = 'Comic Sans MS';  
ax.YLabel.String     = 'Amplitude';  
ax.YLabel.FontSize   = 10;  
ax.YLabel.FontName   = 'Verdana';  
ax.YLabel.Color      = 'b';
```

## Настройка легенды:

```
legend('sin', 'cos', 'Location', 'SouthEast')
```

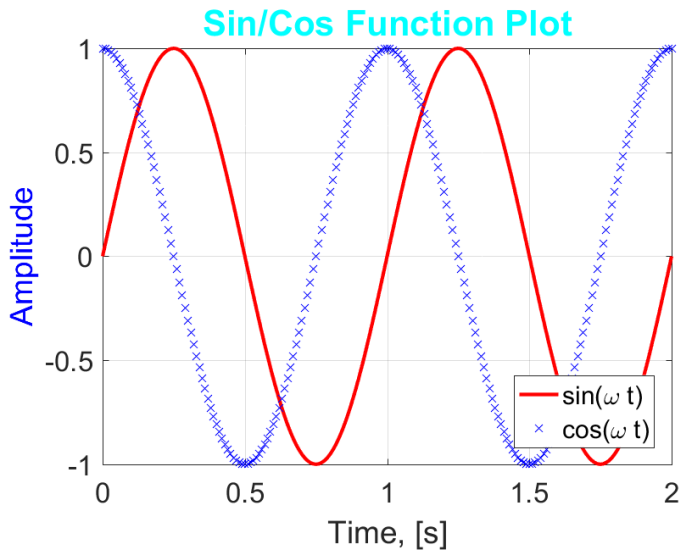
## Настройка легенды через массив строк:

```
legend({'sin'; 'cos'}, 'Location', 'SouthEastOutside')
```

## Использование $\LaTeX$ :

```
legend('sin(\omega t)', 'cos(\omega t)', 'Location', 'SouthEast')
```

# Построение графиков



# Построение графиков

## Виды графиков:

[mathworks.com/help/matlab/creating\\_plots/types-of-matlab-plots.html](https://mathworks.com/help/matlab/creating_plots/types-of-matlab-plots.html)

## Graphics Object Properties:

[mathworks.com/help/matlab/graphics-object-properties.html](https://mathworks.com/help/matlab/graphics-object-properties.html)

## Axes Properties:

[mathworks.com/help/matlab/ref/axes-properties.html](https://mathworks.com/help/matlab/ref/axes-properties.html)

Для каждого вида графика существует свой раздел с настройками

## Chart Line Properties (для `plot`):

[mathworks.com/help/matlab/ref/chartline-properties.html](https://mathworks.com/help/matlab/ref/chartline-properties.html)

## Функции

# Функции

## Синтаксис описания функций:

```
function y = average(x)
    if ~isvector(x)
        error('Input must be a vector')
    end
    y = sum(x)/length(x);
end
```

## Вызов функции:

```
z = 1:99;
c = average(z)
```

Функции сохраняются в .m файлах

Имя файла соответствует имени функции

С версии r2016b года можно сохранять функции в основном скрипте, однако выполнение отдельных секций скрипта становится невозможным



## Функции с несколькими выходными параметрами:

```
function [m,s] = stat(x)
    n = length(x);
    m = sum(x)/n;
    s = sqrt(sum((x-m).^2/n));
end
```

## Вызов функции:

```
values = [12.7, 45.4, 98.9, 26.6, 53.1];
[ave,stdev] = stat(values)
```

Для игнорирования одного из выходных параметров при вызове функции,  
можно на его месте записать ~

### Вложенные функции:

```
function parent
    disp('This is the parent function')
    nestedfx

        function nestedfx
            disp('This is the nested function')
        end
    end
end
```

Вложенные функции могут быть вызваны только из родительской  
Вложенные функции имеют доступ ко всем переменным родительской

### Функция:

```
function y = computeSquare(x)
    y = x.^2;
end
```

### Вызов функции:

```
f = @computeSquare;
a = 4;
b = f(a)
```

Указатели на функцию можно передавать в другие функции  
Можно создавать массив указателей на функции

### Функция:

```
sqr = @(n) n.^2;  
x = sqr(3)
```

У функции нет имени  
Удобны для простых вычислений

## Функции:

[mathworks.com/help/matlab/functions.html](https://mathworks.com/help/matlab/functions.html)






## Об области видимости переменных:

[mathworks.com/help/matlab/variables.html](https://mathworks.com/help/matlab/variables.html)

## Function Precedence Order:

[mathworks.com/help/matlab/matlab\\_prog/function-precedence-order.html](https://mathworks.com/help/matlab/matlab_prog/function-precedence-order.html)

## **Литература**

-  Васильев А.Н.  
*MatLAB. Самоучитель. Практический подход. 2-е издание.*
-  Гаспарян О.Н.  
*MatLAB. Учебное пособие.*
-  Лазарев Ю.Ф.  
*Начала программирования в среде MatLAB.*
-  Мироновский Л.А., Петрова К.Ю.  
*Введение в MatLAB.*
-  Шестаков В.В.  
*Введение в MatLAB. Методические указания к выполнению лабораторных работ по курсу «Компьютерные технологии в приборостроении».*