

```

%%
% Лекция 1. MATLAB
%%
d = 'Think vetorized! - Думай векторно!'
D = 'Think vetorized! - Думай векторно!';
%; -запрет вывода на экран

%% Help
help elfun % элементарные функции
help > % арифметические функции
help matfun % функции со специальными матрицами
% F1

%% Ввод векторов и матриц
% Вектор - строка
a1 = [1 2 3]
% Вектор - строка
a2 = [1, 2, 3]
% Вектор - столбец
a3 = [1; 2; 3]
%%

% Задание матриц
% Матрица, размера 2x3
b1 = [1 2 3; 4 5 6]
% Матрица, размера 3x2
b2 = [1 2; 3 4; 5 6]
% %[нач.знач.:шаг: кон.знач.]
% Задаем вектор x
x = 0:0.01 : 6
% Вычисляем вектор y
y = sin(x)

%% linspace
% Вектор из 100 компонент
c1 = linspace(1, 100)
% Вектор из 5 компонент
c2 = linspace(1, 10, 5)
%% Выделение элементов матрицы
% Очистка экрана
clc
% Очистка переменных
clear
% Задание матрицы
A = [1 2 3; 4 5 6; 7 8 9]
% Изменение 1 - го элемента матрицы
A(1, 1) = 100
% Изменение 3 - й строки матрицы
A(:, 3) = 50
% Изменение 2 - го столбца матрицы
A(2, :) = 33
%%

```

```

% Изменение фрагмента матрицы
clc
% Очистка переменной A
clear A
% Задание квадратной матрицы 7 - го порядка из единиц
A = ones(7)
% Изменение фрагмента матрицы
A(2:6, 2 : 6) = 55
%%

% Использование ключевого слова end
clear A
A = ones(7)
% Изменение фрагмента матрицы
A(4:end, 4 : end) = -21

%% Удаление элементов матрицы
% Задание матрицы
A = [5 5 5; 2 10 2; 2 10 2]
% Удаление 1 - й строки матрицы
A(1, :) = []
% Удаление 2 - го столбца матрицы
A(:, 2) = []
%%

% Удаление нескольких строк матрицы
% Задание матрицы
A = [1 1 1; 2 2 2; 7 3 3]
% Удаление 2 - х строк матрицы
A(1:2, :) = []
% Удаление двух последних элементов матрицы
A(2:end) = []
%%

% Матрица из единиц
% Матрица из единиц 5 - го порядка
A = ones(5)
% Матрица из единиц, в которой 2 строки и 3 столбца
B = ones(2, 3)
%%

% Матрица из нулей 3 - го порядка
C = zeros(3)
% Матрица из нулей, в которой 2 строки и 6 столбцов
D = zeros(2, 6)
%%

% Единичная матрица 3 - го порядка
E = eye(3)
% Единичная матрица, в которой 3 строки и 4 столбца
F = eye(3, 4)
%%

```

```

% Матрица Дюрера 3 - го порядка
M3 = magic(3)
% Матрица Дюрера 5 - го порядка
M5 = magic(5)
%%

% Матрица 5 - го порядка,
% заполненная вещественными случайными числами
% с равномерным распределением из открытого интервала(0, 1)
A = rand(5)
% Матрица 2x3,
A = rand([2 3])
%%

% Используем функцию округления round
% Заполняем матрицу случайными целыми числами от 0 до 10
A = round(rand(8) * 10)
% Заполняем матрицу случайными целыми числами от - 5 до 5
B = round(rand(8) * 10 - 5)
%%

% Поэлементные операции с векторами
% Заполнение векторов. Вектора одинаковой длины!!!
v1 = 10:10 : 50, v2 = 1 : 5
% Поэлементное умножение векторов
r_1 = v1.*v2
% Поэлементное деление векторов
r_2 = v1. / v2
% Поэлементное суммирование векторов и умножение на число
% Точку в данном случае ставить необязательно
r_3 = 0.1*v1 + 100 * v2
%%

% Заполнение матриц. Матрицы одинаковой размерности !!!
m1 = [2 4 6; 3 7 9], m2 = [6 4 2; 9 7 3]
% Поэлементное умножение матриц
z_1 = m1.*m2
% Поэлементное деление матриц
z_2 = m2. / m2
% Поэлементное суммирование матриц и умножение на число
% Точку в данном случае ставить необязательно
z_3 = m1 + 10 * m2
%%

% Заполнение векторов. Векторы одинаковой размерности
q1 = [1 2 3 4], q2 = [10 20 30 40]
% Поэлементное прямое деление векторов
p_1 = q1. / q2
% Поэлементное обратное деление векторов
p_2 = q1.\q2
% Заполнение матриц. Матрицы одинаковой размерности
h1 = [10 20; 30 40], h2 = [5 10; 15 20]
% Поэлементное прямое деление матриц
w_1 = h1. / h2

```

```

% Поэлементное обратное деление матриц
w_2 = h1.\h2
%%

% Матричное умножение
% Задание матриц
M1 = [1 1 1; 2 2 2]
M2 = [3 4; 3 5; 3 6]
% Матричное умножение
M1*M2
% Вектор - строка из 5 элементов
M3 = [1 2 3 4 5]
% Вектор - столбец из 5 элементов
M4 = [1; 2; 3; 4; 5]
% Матричное умножение
M3*M4
% Задание квадратных матриц
%%

%% Матричное обратное деление
clear, clc
% Задание матрицы A и столбца свободных членов b
A = [1 0 0; 0 2 0; 0 0 3]
b = [10; 40; 150]
% Решение системы Ax = b
x = A\b
% или
x = A ^ (-1)*b
% или
x = inv(A)*b
%%

% Возведение матрицы в степень
clear, clc
% Задание матрицы A
A = [1 2 3; 0 2 0; 0 0 3]
% Возведение матрицы в степень
A ^ 2
%%

% Транспонирование вещественной матрицы
clear, clc
A = [1 1 1; 2 2 2; 4 5 6]
% Транспонирование матрицы
A'
% Транспонирование матрицы
A.'
%%

% Транспонирование матрицы, содержащей комплексные элементы
clear, clc
% Задание матрицы A
A = [1 - i 1 + i; 2 + 3i 2 - 3i]

```

```
% Транспонирование матрицы с комплексными значениями
A.'
%%
```

```
% Приоритет матричных операций. Транспонирование и умножение
clear
clc
A = [1 1; 2 2]
% Вычисление значения выражения без скобок
A*A'
% Вычисление значения выражения со скобками
(A*A)'
%%
```

```
% Приоритет матричных операций. Возведение в степень и деление
clear, clc
A = [1 3; 0 5]
% Вычисление значения выражения без скобок
A / A ^ 2
% Вычисление значения выражения со скобками
(A / A) ^ 2
%%
```

```
% Объединение матриц по горизонтали
clear, clc
% Задание матриц
M1 = [1 2; 3 4], M2 = [5 6 7; 8 9 10]
% Объединение по горизонтали с помощью
% квадратных скобок
[M1 M2]
% Объединение по горизонтали с помощью функции cat
cat(2, M1, M2)
%%
```

```
% Объединение матриц по вертикали
clear, clc
% Задание матриц
M3 = [1 2 3], M4 = [5 6 7; 8 9 10]
% Объединение по горизонтали с помощью
% квадратных скобок
[M3; M4]
% Объединение по горизонтали с помощью функции cat
cat(1, M3, M4)
%%
```

```
clear, clc
% Задание матриц
M5 = [1 2; 3 4], M6 = [5 6; 8 9]
% Сложение в «стопку» с помощью функции cat
cat(3, M5, M6)
%%
```

```

% Нахождение обратной матрицы
A = [1 2; 0 2]
inv(A)
A ^ (-1)

%% Часто используемые матричные функции
clear
clc
A = [1 2; 3 4]
sum(A)
%%

% Выделение диагоналей матрицы
clear, clc
A = [1 2 3; 1 2 3; 1 2 3]
% Выделение главной диагонали
diag(A)
% Выделение диагонали, расположенной ниже главной
diag(A, -1)
% Выделение диагонали, расположенной выше главной
diag(A, 1)
%%

% Построение матрицы на основе заданной диагонали
clear, clc
d1 = [1 2 3]
% элементы d1 будут располагаться на главной диагонали
diag(d1)
% элементы d1 будут располагаться ниже
% главной диагонали
diag(d1(2:3), -1)
% элементы d1 будут располагаться выше
% главной диагонали
diag(d1(2:3), 1)
%%

% Функции fliplr и rot90
clc, clear
d1 = [1 2 3], d2 = [11 12 13]
rot90(d1)
fliplr(d2)
a1 = [10 2 3; 40 5 6; 70 8 9]
fliplr(a1)
a2 = [10 20 30; 4 5 6; 7 8 9]
rot90(a2)
%%

% Функция reshape
clc, clear
d = 1:12
size(d)
d = reshape(d, 3, 4)
size(d)
d = reshape(d, 4, [])

```

```

size(d)
%%

% Функции size, numel, length
clc, clear
v1 = [1; 2; 3; 4], v2 = [1 2 3 4]
size(v1), size(v2)
length(v1), length(v2)
numel(v1), numel(v2)
M = [1 2 3; 4 5 6]
size(M), length(M), numel(M)

%% Логические операции с матрицами
% Равно ==
% Не равно ~=
% Больше >
% Больше или равно >=
% Меньше <
    % Меньше или равно <=
    % Логическое И &
    % Логическое ИЛИ |

%% ГРАФИКИ
% Построение графика функции
% задание вектора x
x = [0:0.005 : 5];
% расчет значений функции
y = exp(-x).*sin(10 * x);
% построение графика функции
plot(x, y)
%%

% Два графика функции в одних осях с помощью hold on
x = [0:0.005 : 5];
y1 = exp(-x).*sin(10 * x);
y2 = exp(-x).*cos(10 * x);
% построение первого графика функции
plot(x, y1)
% продолжать построение в этом же окне
hold on
% построение второго графика функции
plot(x, y2)
%%

% Два графика функции в одних осях с помощью plot
x = [0:0.005 : 5];
y1 = exp(-x).*sin(10 * x);
y2 = exp(-x).*cos(10 * x);
% построение сразу двух графиков функций
plot(x, y1, x, y2)
%%

```

```

% Задание цвета и типа линии для графика
x = [0:0.005 : 5];
y = exp(-x).*sin(10 * x);
plot(x, y, 'r:')
%%

% Использование легенды и подписи осей
% создание графического окна
figure
x = [0:0.005 : 5];
y1 = exp(-x).*sin(10 * x);
y2 = sin(10 * x);
% построение графиков двух функций
plot(x, y1, 'k-', x, y2, 'k:')
% создание легенды
legend('y1=exp(-x)*sin(10*x)', 'y2=sin(10*x)', 4)
% создание подписей к осям
xlabel('x')
ylabel('y')
%%

% Использование функции ezplot
% построение графика функции, заданной неявно
ezplot('x^2+(y-abs(x)^(1/2))^2=1')
% построение графика функции, заданной явно
% с заданием отрезка, на котором выполняется построение
ezplot('sin(x)', [0 pi])
%%

% Использование функций axis и text
ezplot('sin(x)')
axis([0 pi 0 1.4])
text(0.25, 1.2, 'text & axis')
%%

% Использование функции subplot
% Гипоциклоиды
clear, clc
% Задание вектора - параметра t
t = 0:0.001 : 8 * pi;
% Данные и вектора для 1 - го окна subplot
k = 5;
x11 = (k - 1)*(cos(t) + cos((k - 1)*t) / (k - 1));
y11 = (k - 1)*(sin(t) - sin((k - 1)*t) / (k - 1));
x12 = k * cos(t);
y12 = k * sin(t);
% Данные и вектора для 2 - го окна subplot
k = 5.5;
x21 = (k - 1)*(cos(t) + cos((k - 1)*t) / (k - 1));
y21 = (k - 1)*(sin(t) - sin((k - 1)*t) / (k - 1));
x22 = k * cos(t);
y22 = k * sin(t);
% Данные и вектора для 3 - го окна subplot
k = 6;

```



```

x31 = (k - 1)*(cos(t) + cos((k - 1)*t) / (k - 1));
y31 = (k - 1)*(sin(t) - sin((k - 1)*t) / (k - 1));
x32 = k * cos(t);
y32 = k * sin(t);
% Данные и вектора для 4 - го окна subplot
k = 3.6;
x41 = (k - 1)*(cos(t) + cos((k - 1)*t) / (k - 1));
y41 = (k - 1)*(sin(t) - sin((k - 1)*t) / (k - 1));
x42 = k * cos(t);
y42 = k * sin(t);
% Построение графиков в 1 - м окне subplot
subplot(2, 2, 1)
plot(x11, y11, x12, y12)
legend('k=5')
% Построение графиков во 2 - м окне subplot
subplot(2, 2, 2)
plot(x21, y21, x22, y22)
legend('k=5.5')
% Построение графиков в 3 - м окне subplot
subplot(2, 2, 3)
plot(x31, y31, x32, y32)
legend('k=6')
% Построение графиков в 4 - м окне subplot
subplot(2, 2, 4)
plot(x41, y41, x42, y42)
legend('k=3.6')
% %

```

```

% Использование логарифмической шкалы для оси x
clear, clc
x = [1e+0, 1e+1, 1e+2, 1e+3, 1e+4, 1e+5, 1e+6]
y = [1, 2, 4, 8, 16, 32, 64];
p = semilogx(x, y), set(p, 'LineWidth', 2);
title('Логарифмическая шкала по оси x')
% %

```

```

clear, clc
x = [1e+0, 1e+1, 1e+2, 1e+3, 1e+4, 1e+5, 1e+6]
y = [1e+1, 1e+4, 1e+1, 1e+3, 1e+1, 1e+4, 1e+1];
p = loglog(x, y)
set(p, 'LineWidth', 2);
title('Логарифмическая шкала для обеих осей')
grid on % задание сетки
%%

```

```

% График в полярных координатах
theta = 0:0.01 : 2 * pi;
rho = sin(2 * theta).*cos(2 * theta);
figure
polar(theta, rho, '--r')
% %

```